

Joona Hirvasmaa

**KÄYTTÄJÄKESKEISEN SUUNNITTELUN
YHDISTÄMINEN KETTERÄÄN
OHJELMISTOKEHITYKSEEN**
Suunnittelijan näkökulma

Informaatioteknologian ja viestinnän tiedekunta
Pro gradu -tutkielma
Lokakuu 2019

TIIVISTELMÄ

Joona Hirvasmaa: Käyttäjäkeskeisen suunnittelun yhdistäminen ketterään ohjelmistokehitykseen: Suunnittelijan näkökulma
Pro gradu -tutkielma
Tampereen yliopisto
Tietojenkäsittelytieteiden tutkinto-ohjelma
Lokakuu 2019

Ketterä kehitys kuuluu suosituimpiin ohjelmistokehitysmenetelmiin, mutta menetelmässä käyttäjäkeskeisyys jää usein taka-alalle, jolloin ohjelmiston käytettävyys ja käyttäjäkokemus kärsii. Tämän tutkielman tarkoituksena olikin selvittää, miten käyttäjäkeskeinen suunnittelu ja ketterä kehitys voidaan integroida, jotta voidaan kehittää laadukkaita ohjelmistoja ketterästi. Tämä tutkimus muodostuu tehdystä kirjallisuuskatsauksesta sekä empiirisestä tutkimuksesta. Kirjallisuuskatsauksessa selvitettiin tieteellisistä tutkimuksista, miten käyttäjäkeskeistä suunnittelua ja ketterää kehitystä yhdistetään ohjelmistotalalla. Kirjallisuuskatsauksen tulosten pohjalta luotiin empiirisen haastattelututkimuksen teemat ja näiden haastattelukysymykset. Haastatteluihin osallistui kahdeksan suunnittelijaa, joiden työnkuvaan kuului käyttäjäkeskeinen suunnittelu ketterässä ohjelmistokehityksessä.

Tehdyn empiirisen tutkimuksen tulokset olivat suurilta osin samassa linjassa kirjallisuuskatsauksen kanssa, mutta myös mielenkiintoisia eroavaisuuksia löytyi. Suurimmat erot ilmenivät ketterän menetelmän käytössä ja suunnittelutoimien suhteesta ohjelmistoprojektien rahoitukseen. Keskeiseksi tulokseksi selvisi suunnitteluratkaisujen kommunikoinnin tärkeys. Muihin tuloksiin lukeutui muokatun Scrum-menetelmän käyttäminen, ketteryyden korostaminen suunnittelussa, suunnittelun ja kehityksen tiivis yhdessä kulkeminen, suunnittelutoimien pitäminen kehityksen edellä, kahden tai useamman suunnittelijan käyttäminen omine rooleineen, esivaiheen käyttäjätutkimuksen painottaminen, prototyyppien korostaminen, loppukäyttäjän aktiivinen osallistaminen ja asiakkaan kanssa tiiviin yhteistyön ylläpitäminen.

Tämän tutkimuksen tulosten pohjalta pystyttiin luomaan kaksi integrointimallia, joiden avulla käyttäjäkeskeinen suunnittelu ja ketterä kehitys voidaan yhdistää. Ennen kaikkea tutkielman tuloksena saatiin listaus käytänteitä ja menetelmiä, mitkä vaikuttavat onnistuneen käyttäjäkeskeisen suunnittelun ja ketterän kehityksen yhdistämiseen.

Avainsanat: käyttäjäkeskeinen suunnittelu, käyttäjäkokemus, ketterä kehitys, ketteryys, Scrum, Lean, Kanban, ohjelmistokehitys

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

Esipuhe

Olen saanut opiskelut päätökseen Pro gradu -tutkielman valmistumisen myötä ja pitkään odotettu valmistuminen odottaa. Tämän tutkielman kirjoittaminen on ollut pitkä prosessi ja haluankin kiittää henkilöitä, jotka ovat auttaneet ja tukeneet minua tämän prosessin ajan.

Haluan ensin kiittää kaikkia tutkielmani haastatteluihin osallistuneita ohjelmistoyritysten suunnittelijoita. Ilman heitä tämä tutkimus ei olisi onnistunut. Haluan kiittää myös tutkielmani ohjaajaa Tarja Tiaista asiantuntevasta opastuksesta, kannustavasta asenteesta sekä mielenkiinnosta tutkielmani aihetta kohtaan. Lisäksi haluan kiittää perhettäni sekä kavereitani, joiden tuki ja kannustus opintojeni aikana oli korvaamaton.

Lopuksi haluan kiittää vielä erikseen opiskelijakavereitani, joiden kanssa opiskeleminen sekä vapaa-ajan viettäminen auttoivat jaksamaan ja tekivät opiskeluvuosistani ikimuistoiset. Opiskeluajoiltani jääkin lämpimät muistot, joihin on aina mukava palata.

Tampereella 16.10.2019

Joona Hirvasmaa

Sisällysluettelo

1. Johdanto	1
2. Ketterän kehityksen ja käyttäjäkeskeisen suunnittelun keskeiset käsitteet	4
2.1 Ohjelmistokehitys	4
2.2 Ketterä ohjelmistokehitys	5
2.2.1 Scrum	8
2.2.2 Lean	11
2.2.3 Kanban	14
2.3 Käyttäjäkeskeinen suunnittelu ohjelmistokehityksessä	15
2.3.1 Prosessi	19
2.3.2 Käyttäjäkeskeinen lähestymistapa	21
2.3.3 Toimintatavat ja välineet käyttäjäkeskeisessä suunnittelussa	22
3. Suunnittelun ja ketteryyden integroinnin kirjallisuuskatsaus	26
3.1 Tietokantahaut	26
3.2 Ketteryyden ja käyttäjäkeskeisen suunnittelun yhdistäminen	28
3.3 Käytänteet ja menetelmät mallien yhdistämiseksi	29
3.3.1 Integroitu prosessi	30
3.3.2 Etukäteen suunnittelu	33
3.3.3 Käyttäjäkeskeinen suunnittelija osana kehitystiimiä	34
3.3.4 Välineet ja toimintatavat suunnittelussa	38
3.3.5 Sidosryhmien mukana pitäminen	40
3.4 Kirjallisuuskatsauksen yhteenveto	41
4. Tutkimusmenetelmät	45
4.1 Laadullinen tutkimus	45
4.2 Aineiston keruumenetelmä	46
4.3 Haastattelusuunnitelma	47
4.3.1 Haastateltavat	47
4.3.2 Haastatteluteemat	48
4.3.3 Haastatteluprosessi	49
4.4 Haastattelutoteuma	50
4.5 Haastattelutulosten analysointi	51
5. Haastattelutulokset	53
5.1 Teemat	53
5.1.1 Integroitu prosessi	54
5.1.2 Suunnittelija osana kehitystiimiä	58
5.1.3 Etukäteen suunnitteleminen	63
5.1.4 Kommunikointi suunnitteluratkaisuista	68
5.1.5 Suunnittelun toimintatavat ja välineet	73
5.1.6 Sidosryhmät	76
5.2 Tyypit	79
5.2.1 Suunnittelupainotteinen integraatiomalli	79
5.2.2 Tasapainotettu integraatiomalli	82
5.2.3 Integraatiomallien yhteenveto	83
6. Pohdinta ja johtopäätökset	85
6.1 Tutkimustulokset ja kirjallisuuskatsaus	85
6.2 Keskeiset johtopäätökset	91
6.3 Jatkotutkimusaiheet	93
6.4 Tämän tutkimusprosessin arviointi	94

6.5	Tämän tutkimuksen luotettavuus.....	96
6.6	Tämän tutkimuksen rajoitteet	98
7.	Yhteenveto.....	99
8.	Viiteluettelo	100
9.	Liiteluettelo	105
	Liite 1: Kirjallisuuskatsaukseen valittu aineisto.....	105
	Liite 2: Tutkimuksen laadun arviointi	107
	Liite 3: Haastattelukutsu.....	108
	Liite 4: Haastattelurunko teemoineen.....	109

1. Johdanto

Kysymys siitä, kuinka ohjelmistokehityksen haasteisiin pitäisi vastata on ollut esillä tutkijoiden piirissä siitä asti, kun ohjelmistokehitys nousi omaksi tieteen koulukunnaksi 1970-luvulla. Nykyään ohjelmistokehityksessä pyritään ymmärtämään kehitysaktiviteetteja dynaamisina prosesseina, joita kuvataan iteraatiokierroksilla ja aktiivisella sidosryhmien osallistamisella. Tämä kuvastaa ketterää ohjelmistokehitystä, jossa vastataan arvaamattomiin muutoksiin ihmisten ja heidän luovuuden avulla eikä prosessien. Ohjelmistokehitysaktiviteetit rajoitetaan myös niihin, mitkä luovat arvoa asiakkaalle. [Brhel et al., 2015, s. 164]

Ketterän kehityksen tuomat edut ovat tehneet siitä yleisimmän ohjelmistokehitysmenetelmän [Jurca et al., 2014, s. 24]. Ketterä kehittäminen on yleistynyt jopa isoissa ohjelmistoyrityksissä. Ketteryyden avulla ohjelmistoja pystytään toimittamaan ajallaan, budjetin puitteissa ja asiakkaan vaatimuksiin voidaan vastata aina paremmin. Ketterät menetelmät eivät kuitenkaan keskity siihen, kuinka hyödyllinen (eng. useful) ohjelmisto voidaan kehittää tai siihen onko se käytettävä (eng. usable) eli sisältää hyvän käytettävyyden. Hyvän käytettävyyden omaavista ohjelmistoista on puolestaan tullut tärkeitä erityisesti, koska niillä pystytään luomaan kilpailuetua kasvavilla markkinoilla. [Brhel et al., 2015, s. 164]

Käyttäjäkeskeinen suunnittelu asettaa loppukäyttäjien tavoitteet ja tarpeet ohjelmistokehityksen keskiöön, jotta pystytään kehittämään ohjelmisto hyvällä käytettävyydellä. Yhdistämällä ketterän kehityksen sekä käyttäjäkeskeisen suunnittelun pyritään saavuttamaan ohjelmisto, joka on hyödyllinen ja käytettävä. [Brhel et al., 2015, s. 163] Käyttäjäkeskeisen suunnittelun tavoitteena on parantaa käytettävyyttä siihen pisteeseen, että käyttäjä pystyy navigoimaan käyttöliittymässä helposti. Jos ohjelmistolla on huono käytettävyys, sen käyttämisen tehokkuus laskee tai käyttäjä voi lopettaa ohjelmiston käytön kokonaan. Näistä syistä monet ohjelmistoyritykset ovat alkaneet etsiä ratkaisuja yhdistää käytettävyyssperiaatteita ketterään kehitykseen. [Jurca et al., 2014, s. 24]

Tällä tutkimuksella pyritään selvittämään, kuinka käyttäjäkeskeinen suunnittelu ja ketterä kehittäminen voivat hyötyä toisistaan. Lisäksi selvitetään, miten ne voidaan yhdistää. Menetelmien yhdistämisessä on haasteensa, mutta yhdistäminen voidaan nähdä kannattavana, koska käyttäjäkeskeinen suunnittelu tuo ketteryyteen puuttuvan käytettävyystietoisuuden ja käyttäjäkeskeisyyden. Aihe on oleellinen ohjelmistoyrityksien edustajien ja tutkijoiden kannalta. Ohjelmistoyritysten edustajille tämä tutkimus tarjoaa selvityksen hyvistä käytännöistä ja menetelmistä yhdistää käyttäjäkeskeinen suunnittelu sekä ketterä kehitys. Tutkijoille tämä tutkimus tarjoaa kirjallisuuskatsauksen ja haastattelututkimuksen, jotka osoittavat aukot nykyisissä tutkimuksissa, joita pitäisi tutkia tarkemmin.

Samalla vahvistetaan nykyisiä tutkimustuloksia ja eroavaisuudet tuodaan esille. Tämän tutkimuksen tutkimuskysymyksenä on:

Kuinka käyttäjäkeskeinen suunnittelu ja ketterä kehitys voidaan yhdistää keskenään ohjelmistokehityksessä suunnittelijan näkökulmasta.

Tutkimuskysymykseen vastattiin tehdyllä kirjallisuuskatsauksella aiemmista aiheeseen liittyvistä tutkimuksista sekä tässä tutkimuksessa tehdyn empiirisen haastattelututkimuksen avulla. Aiheen laajuuden takia, se on rajattu käyttäjäkeskeisen suunnittelun näkökulmaan ja ennen kaikkea suunnittelijan näkökulmaan. Kahden menetelmän integrointi on laaja, joten oli perusteltua rajata se tiettyyn näkökulmaan. Pelkästään tutkimuskysymyksen kannalta itse suunnittelijoiden näkemyksien ja mielipiteiden selvittäminen on tärkeää, koska integrointi koskee pitkälti suunnittelijoiden tekemää työtä ohjelmistokehityksessä. Suunnittelijoita haastattelemalla voidaan selvittää hyväksi koetut käytänteet ja mahdolliset ongelmat integraatiossa.

Käyttäjäkeskeisen suunnittelun ja ketterän kehityksen yhdistämisestä on tehty jo monia eri tutkimuksia, mutta niissä ei pureuduta suoraan onnistuneen yhdistämisen aikaansaamiseen, vaan enemmän ongelmien luettelemiseen. Tutkimukset painottavat enemmän helpotuksia integraation onnistumiselle kuin selviä ratkaisuja. Käyttäjäkeskeisyyden kannalta selvästi suunnittelijoiden näkökulmasta tehdyt tutkimukset ovat harvassa. Suunnittelijan näkökulmaa voidaan pitää tärkeänä, koska itse käyttäjäkeskeinen suunnittelu koskee suunnittelijoiden tekemää työtä osana ohjelmistokehitystä aivan kuten edellä todettiin. Suomessakaan ei ole tehty paljoa tutkimusta aiheen tiimoilta, joten tällä tutkimuksella pyritään selvittämään myös käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integroinnin nykytilaa Suomen ohjelmistoyrityksissä. Tieteellisten tutkimusten lisäksi aiheesta on mahdollista lukea erilaisista verkkojulkaisuista sekä erityisesti ohjelmistoyritysten omista blogikirjoituksista. Aihe on yhä oleellinen, koska se on ollut pinnalla jo vuosia ja siitä puhutaan edelleen.

Tutkimuksen alussa luvussa 2 kerrotaan ensin keskeiset käsitteet. Luvussa käsitellään ohjelmistokehitys, ketterä kehitys ja käyttäjäkeskeinen suunnittelu.

Luvussa 3 on kirjallisuuskatsaus, jossa esitellään aiheeseen liittyvien muiden tutkimuksien löydökset. Luvussa luodaan kokonaiskuva aiemmista tutkimuksista ja näiden pohjalta luodaan tämän tutkimuksen empiirisen tutkimuksen pohja.

Luvussa 4 kerrotaan tämän tutkimuksen tutkimusmenetelmät. Luvussa pohditaan, mitkä aineistonkeruumenetelmistä sopii parhaiten tähän tutkimukseen ja valitaan sopivat

menetelmät. Luvussa käsitellään lisäksi haastattelusuunnitelma, haastattelutoteuma ja alustetaan haastattelujen analysointi.

Luvussa 5 kerrotaan haastattelutulokset. Haastattelutulokset käsitellään ensin teemoittelun ja tämän jälkeen tyypittelyn avulla. Teemoittelun avulla luodaan teemoja eli keskeisiä aiheita etsimällä aineistosta yhdistäviä ja erottavia seikkoja. Tyypittelyssä puolestaan aineisto tiivistetään havainnollistaviin tyyppeihin. Tulokset on tuotu näkyviksi teemojen avulla ja näitä on verrattu myös kirjallisuuskatsauksen tuloksiin. Syntyneiden teemojen pohjalta ne on vielä tyypitelty omiin tyyppeihin, joilla tulokset jäsennetään omiin kokonaisuuksiin.

Luvussa 6 pohditaan tätä tutkimusta kokonaisvaltaisesti. Tutkimustuloksien yhteyttä kirjallisuuteen pohditaan sekä katsotaan koko tutkimusprosessia ja sen onnistumista. Luvussa esitetään lisäksi tämän tutkimuksen keskeiset johtopäätökset sekä mahdolliset jatkotutkimusaiheet. Lopuksi luvussa pohditaan tämän tutkimuksen luotettavuutta ja mahdollisia rajoitteita.

Luvussa 7 esitellään tutkimuksen tärkeimmät löydökset. Luvussa on lyhyt yhteenveto koko työstä.

2. Ketterän kehityksen ja käyttäjäkeskeisen suunnittelun keskeiset käsitteet

Tieteellisessä tiedonhankinnassa pyritään aina käsitteellistämään tutkittavaa ilmiötä. Tällä tavalla pyritään hahmottamaan asioita yleisellä sekä teoreettisella tasolla. Lähtökohtana on, että tutkimus rakentuu aina käsitteellisten merkitystulkintojen ja ratkaisujen varaan. [Hirsjärvi et al., 2007, s. 142-143]

Tärkeimmät tutkimuksessa käytettävät käsitteet on määriteltävä selvästi [Hirsjärvi et al., 2007, s. 147]. Tälle tutkimuksella tärkeimmät käsitteet ovat: ohjelmistokehitys, ketterä ohjelmistokehitys ja käyttäjäkeskeisen suunnittelu ohjelmistokehityksessä. Näistä jokainen käsitellään omissa luvuissaan.

2.1 Ohjelmistokehitys

Ohjelmistot ovat osa modernia yhteiskuntaa. Ne ovat abstrakteja ja aineettomia. Ohjelmistot eivät ole ainoastaan tietokoneohjelmia itsessään vaan niihin sisältyy myös dokumentointia ja tiedon konfiguraatiota, jotta ohjelmistot toimivat oikein. Ohjelmistoja on paljon eri tyyppisiä aina yksinkertaisista monimutkaisiin ja yksi ohjelmisto voi koostua useista eri ohjelmistoista. Ohjelmistoja kehitetään johonkin tarpeeseen ja yleensä niitä kehitetään tietylle asiakkaalle tai yleisille markkinoille. [Sommerville, 2011, s. 4-6]

Itse ohjelmistokehitys kattaa kaiken toiminnan siitä, miten ohjelmisto kehitetään aina ohjelmiston vaatimusten keräämisestä sen ylläpitoon. Ohjelmiston kehityksessä tavoitteena on saada määritellyn laadun omaava ohjelmisto valmiiksi tietyssä aikataulussa ja budjetissa. Tämän takia ohjelmistokehityksessä joudutaan usein tekemään kompromisseja. Rajoitusten takia kehitykseen tulee valita sopivimmat tavat. Ohjelmistokehitys on oma prosessinsa, jota voidaan yksinkertaisesti kutsua ohjelmistoprosessiksi. Ohjelmistoprosessi on joukko aktiviteetteja, jotka vievät ohjelmistokehitystä eteenpäin aina valmiiseen ohjelmistoon. On olemassa neljä pääaktiviteettia, jotka ovat yleisiä jokaisessa ohjelmistoprosessissa. [Sommerville, 2011, s. 7-9] Nämä aktiviteetit ovat [Sommerville, 2011, s. 9]:

1. Ohjelmistovaatimukset: Asiakas ja ohjelmistokehittäjät määrittelevät ohjelmiston vaatimukset ja mahdolliset rajoitteet.
2. Ohjelmistokehitys: Ohjelmisto suunnitellaan ja kehitetään.
3. Ohjelmistovalidointi: Varmistetaan että ohjelmisto vastaa asiakkaan asettamia vaatimuksia.
4. Ohjelmistomuutos: Ohjelmistoa muokataan asiakkaan ja markkinoiden vaatimuksien mukaan.

Ohjelmistoprosessit voidaan jakaa suunnitelmapainotteisiin- tai ketteriin prosesseihin. [Sommerville, 2011, s. 29] Tämä tutkielma nivoutuu ketterään kehitykseen ja sitä käsitellään seuraavassa kohdassa.

Ohjelmistokehitys sisältää monia käsitteitä. Tässä tutkielmassa ohjelmistolla tai tuotteella tarkoitetaan ohjelmistokehityksessä syntyvää lopputuotetta. Sanalla suunnittelija tarkoitetaan henkilöä, jonka työnkuviin kuuluu käyttäjäkeskeinen suunnittelu ohjelmistoprojekteissa, jollei toisin todeta. Kehittäjällä ja toteuttajalla puolestaan tarkoitetaan henkilöä, joka vastaa ohjelmiston ohjelmoinnista.

2.2 Ketterä ohjelmistokehitys

Monissa eri ohjelmistokehitysmenetelmissä kehittäminen aloitetaan vaatimusten dokumentoinnilla. 1990-luvun puolessa välissä moni havaitsi tämän varhaisen dokumentaation turhauttavana ja jopa mahdottomana. Teknologia- ja liiketoimintaympäristöt muuttuivat projektien aikana nopeasti sekä vaatimukset ja projektisuunnitelmat vanhenivat lyhyissä projekteissa. Asiakkaille tuli myös koko ajan haastavammaksi määritellä vaatimuksia. Näistä syistä ohjelmistoalan ketterän kehityksen puolestapuhujat alkoivat kehittää menetelmiä ja käytäntöjä, jotka kannustaisivat muutokseen. Menetelmiä kehitettiin Euroopassa, Australiassa ja Yhdysvalloissa. Näihin luotuihin menetelmiin kuuluivat muun muassa Extreme Programming ja Scrum. [Williams ja Cockburn, 2003, s. 39]

Vuonna 2001 helmikuussa 17 näiden uusien menetelmien kehittäjistä tapasivat Snowbird Utah:ssa, jossa he keskustelivat menetelmiensä samankaltaisuuksista ja niin sanotuista kevyistä menetelmistä. Nopeasti kävi ilmi, että heidän toimintatavoissa oli samankaltaisuuksia ja näillä kevyillä menetelmillä saavutettiin käyttäjätyytyväisyyttä sekä korkeaa laatua. He kategorisoivat menetelmät ”ketterä”-termin alle. Tämän tapaamisen merkittävänä tuloksena oli ketterän manifestin (eng. Agile Manifesto) luominen, jota pidetään yleisesti ketterän kehityksen perusmääritelmänä. [Williams ja Cockburn, 2003, s. 39]

Ketterän manifestin tarkoituksena on luoda yhtenäiset menetelmät, joita ketterien menetelmien tulisi noudattaa. Manifesti määrittää ketteryyden neljä perusmenetelmää sekä 12 periaatetta, joita nämä menetelmät noudattavat. Nämä neljä menetelmää koostuvat seuraavista [Agile Alliance, 2001]:

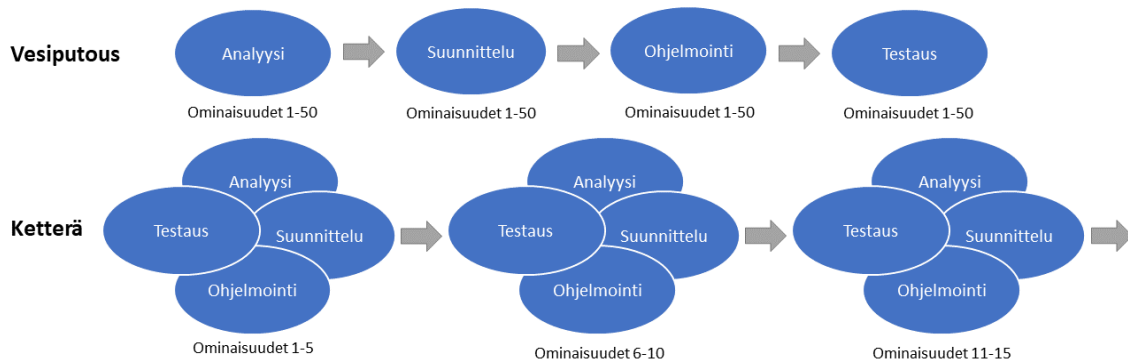
1. Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja.
2. Toimiva ohjelmisto ennen kattavaa dokumentointia.
3. Asiakasyhteistyö ennen sopimusneuvotteluja.
4. Muutoksiin vastaamista enemmän kuin suunnitelmissa pitäytymistä.

Näiden menetelmien lisäksi ketterä manifesti koostuu 12 periaatteesta, joita ketteryyss noudattaa [Agile Alliance, 2001]:

1. Tärkein tavoite on asiakkaan tyytyväisyyden takaaminen, toimittamalla nopeasti ja säännöllisesti uusia versioita ohjelmistosta.
2. Muuttuvat vaatimukset otetaan vastaan myös kehityksen myöhäisessä vaiheessa. Ketteryys hyödyntää muutosta asiakkaan kilpailukyvyn lisäämiseksi.
3. Versioita toimivasta ohjelmistosta toimitetaan säännöllisesti, muutaman viikon tai kuukauden välein. Lyhyempää aikaväliä suositaan.
4. Ohjelmistokehittäjien ja liiketoiminnan edustajien tulee työskennellä yhdessä päivittäin koko projektin aikana.
5. Projekti rakennetaan motivoituneiden henkilöiden ympärille ja heille annetaan puitteet ja tuki, jonka he tarvitsevat. Henkilöihin tulee luottaa, jotta he saavat projektin tehtyä.
6. Kasvokkain käydyt keskustelut ovat tehokkain ja toimivin tapa tiedon välittämiseksi kehitystiimille ja tiimin jäsenten kesken.
7. Toimiva ohjelmisto toimii ensisijaisena edistymisen mittarina.
8. Ketterät menetelmät suosivat kestäväää toimintatapaa. Hankkeen omistajien, kehittäjien ja lopullisen ohjelmiston käyttäjien tulisi pystyä ylläpitämään työtah- tinsa tulevaisuudessakin.
9. Ketteryyttä edesauttaa teknisen laadun ja ohjelmiston hyvän rakenteen jatkuva huomiointi.
10. Oleellista on yksinkertaisuus ja tekemättä jätettävän työn maksimointi
11. Parhaat suunnitelmat, arkkitehtuurit ja vaatimukset syntyvät itseorganisoituvissa kehitystiimeissä.
12. Kehitystiimi tarkastelee säännöllisesti, kuinka tehokkuutta voisi parantaa ja muokata toimintaa sen mukaisesti.

Ketterän manifestin esittämät menetelmät ja periaatteet ovat enemmän suuntaa antavia, koska ne eivät määrittele varsinaisesti ketteryyttä. Abrahamsson et al. [2002] tiivistävät ketterät menetelmät yksinkertaisuuteen ja nopeuteen. Kehityksessä kehitystiimi keskittyy vain ensikädessä tarvittaviin toimintoihin ja toimittaa ne nopeasti, samalla keräten pa- lautetta ja reagoiden uuteen informaatioon. Ohjelmistokehitys on ketterää, kun kehitys on inkrementaalista (pieniä ohjelmistojulkaisuja nopeissa sykleissä), yhteistyö pohjaista (asiakas ja kehittäjät työskentelee yhdessä jatkuvasti), suoraviivaista (menetelmä on helppo oppia, muokata ja dokumentoida) ja adaptoituvaa (mahdollisuus tehdä viime het- ken muutoksia). [Abrahamsson et al., 2002, s. 19]

Nämä inkrementaaliset julkaisut sisältävät suunnittelun, koodauksen, testauksen ja analysoinnin. Jokaisen julkaisun lopputuloksena on toimiva ohjelmistoversio, joka sisältää tietyn ennalta sovitun määrän kehitettyjä ominaisuuksia. Ketterä kehitys nähdäänkin vastakohtana perinteiselle vesiputousmallille, jossa kehitysvaiheet tulevat peräkkäin (Kuva 1). [Sy, 2007, s. 113-114]



Kuva 1. Vesiputousmalli ja ketterä kehitys. Kuva muokattu lähteestä [Sy, 2007, s. 114].

Highsmithin ja Cockburnin [2001, s121] mukaan ketteryydessä painotetaan kahta eri konseptia, jotka ovat toimiva koodi ja työskentelevien ihmisten tehokkuus. Toimiva koodi kertoo sen, mitä asiakas loppujen lopuksi saa, eikä se perustu vain lupauksiin. Ihmisten tehokkaalla käytöllä puolestaan saavutetaan ohjattavuutta, nopeutta ja säästöjä kuluissa. Ihmiset voivat vaihtaa ajatuksiaan ja ideoitaan paremmin kasvokkain tapaamisissa, kuin dokumentteja lukemalla. Miller [2001] puolestaan kiteyttää ketteryyden yhdeksään eri tyyppiin: modulaarisuus, iteratiivisuus, aikasidonnaisuus, säästäväisyys, adaptoitavuus, inkrementaalisuus, konvergenssisuus, ihmiskeskeisyys ja yhteistyö. Loppujen lopuksi ketteryys on muutoksen luomista ja siihen vastaamista. Ketteryydessä esiintyvät käytänteet eivät ole uusia, vaan ihmisten tunnistaminen avaintekijöiksi projektin onnistumiseksi. Näiden lisäksi keskitytään tehokkuuteen ja ohjattavuuteen. [Highsmith ja Cockburn, 2001, s. 122]

Ketterät menetelmät ovat saavuttaneet suosiota viime vuosina perinteisten ohjelmistokehitysmenetelmien rinnalle. Ketterään ohjelmistokehitykseen on esitetty useita eri menetelmiä. Yleisimmät näistä menetelmistä ovat: Scrum, Lean, Extreme programming, kristalli menetelmä, Dynamic Systems Development Method ja Feature-Driven-Development [Dybå ja Dingsøyr, 2008, s. 835]. Menetelmien laajuuden vuoksi niitä kaikkia ei voida käsitellä, joten karsintaa on tehtävä. VersionOne niminen yhdysvaltalainen yritys toteuttaa joka vuosi laajan kyselyn maailmanlaajuisesti ketterien menetelmien käytöstä. Uusimman tutkimuksen mukaan selvästi yleisin menetelmä on Scrum, jota käyttää 56 % vastaajista. Kanban menetelmää käytti 5 % ja Lean menetelmää vain 1 %. Näiden mallien hybridimenetelmät olivat myös hyvin käytössä: Scrum/XP 6 %, mukautettu hybridimalli

14 % ja Scrumban 8 %. [VersionOne, 2018, s. 9] Suomessakin on tutkittu ketterien menetelmien käyttöä, kun Rodríguez et al. [2012] teettivät Suomessa vuonna 2011 kyselyn ohjelmistoyrityksille ketterien menetelmien käytöstä. Tutkimuksessa kerrottiin, että suosituimpia olivat Scrum, Extreme Programming, Agile Modeling, Feature-Driven Development ja Kanban. [Rodríguez et al., 2012] Aikaisemmat tutkimukset tukevat tässä tutkielmassa tehdyn empiirisen tutkimuksen tuloksia, koska käytetyimmiksi menetelmiksi kerrottiin myös Scrum ja Kanban. Lisäksi Lean kuului käytettyihin menetelmiin, jonka käyttö puolestaan aikaisempien tutkimuksien perusteella oli vähäistä. Seuraavaksi luodaan katsaus tämän tutkimuksen kannalta tärkeisiin ketteriin menetelmiin eli Scrumiin, Leaniin ja Kanbaniin.

2.2.1 Scrum

Scrum-menetelmän viitekehys perustuu Scrum kehittäjien Schwaberin ja Sutherlandin [2017] ajatuksiin. Heidän mukaan Scrumin avulla ihmiset voivat ratkaista monimutkaisia ongelmia ja samalla kehittää huomattavaa lisäarvoa tarjoavia tuotteita luovasti ja tuottavasti. Scrum koostuu iteraatioista (eng. iteration) eli kehityssykleistä, joissa suunnittelu ja toteutus tehdään pienissä osissa kerrallaan ja tätä prosessia toistetaan. Tuote kehittyy aina täydemmäksi jokaisen kehityskierroksen jälkeen. [Sutherland, 2010, s. 6] Tiivistetynä Scrum on kevyt ja yksinkertainen ymmärtää, mutta vaikea taitaa. Scrum on viitekehys, jonka sisällä on mahdollista hyödyntää monia eri prosesseja ja tekniikoita. Menetelmällä tuotteenhallinta ja työskentelymenetelmien suorituskky tulee näkyviksi, jotta tuotetta, tiimiä ja työskentely-ympäristöä voidaan parantaa jatkuvasti. Viitekehys pitää sisällään Scrum-tiimin rooleineen, tapahtumat, tuotokset ja säännöt. [Schwaber ja Sutherland, 2017, s 3]

Schwaberin ja Sutherlandin [2017, s. 5] mukaan: ”Scrumin arvot ovat sitoutuminen, rohkeus, keskittyminen, avoimuus ja kunnioitus”. Kun näiden arvojen mukaan toimitaan, niin Scrum-tiimi vahvistaa Scrumin kolmea tukijalkaa, jotka ovat läpinäkyvyys, tarkastelu ja sopeutuminen. Nämä puolestaan kasvattavat luottamusta. Scrum-menetelmän onnistuminen riippuu siitä, kuinka hyvin ihmiset pystyvät omaksumaan näitä Scrumin viittä arvoa. Jokainen Scrum-tiimin jäsen sitoutuu tavoitteisiin ja keskittyy sprintin (eng. sprint) työhön. Jäsenet kunnioittavat toisiaan itsenäisinä ja kyvykkäinä ihmisinä. [Schwaber ja Sutherland, 2017, s. 5]

Scrum-tiimi koostuu kolmesta eri roolista, jotka ovat tuotteen omistaja (eng. product owner), kehitystiimi (eng. development team) ja scrummaster (eng. scrum master). Scrum-tiimit ovat monitaitoisia ja itseohjautuvia. Tiimit päättävät itse, kuinka tekevät työnsä, eikä ulkoista ohjausta ole. Tämä tiimimalli on suunniteltu luovuuden, joustavuuden ja tuottavuuden optimoimiseksi. Tiimit toimittavat tuotetta jatkuvasti ja lisäävällä

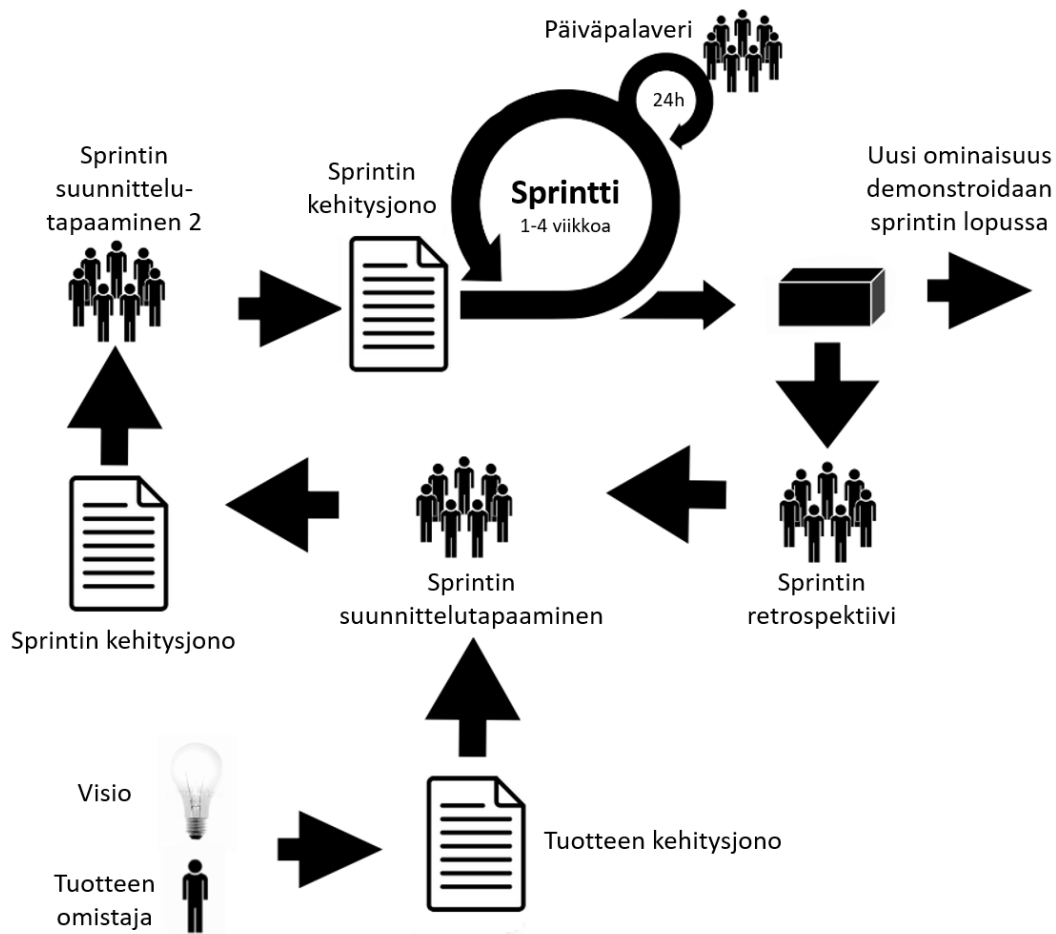
määrällä, näin maksimoiden palautteen saamisen mahdollisuudet. [Schwaber ja Sutherland, 2017, s. 6]

Tuotteen omistaja on päävastuussa tuotteen arvon maksimoinnista, joka on tulos kehitystiimin työn tuloksesta [Schwaber ja Sutherland, 2017, s. 6]. Tuotteen omistaja konkretisoi visionsa tuotteesta vaatimuksiksi, jotka kirjataan ja priorisoidaan tuotteen kehitysjonoon (eng. product backlog). Tuotteen omistaja vastaakin tuotteen kehitysjonosta. Tuotteen omistaja päättää, mitä ominaisuuksia tuotteeseen rakennetaan ja missä järjestyksessä. Tuotteen omistaja on tuotekehityksen johtavassa roolissa. Tuotteen omistaja on mukana suunnittelussa ja kommunikoi kehitystiimin sekä asiakkaan kanssa. [Sutherland, 2010, s. 10-14]

Kehitystiimi on oleellisessa osassa Scrum-tiimiä, koska se vastaa tuotteen kehitysjonon sisällön muuntamista julkaisukelpoiseksi inkrementiksi jokaisessa sprintissä. Tiimit ovat itseohjautuvia ja kukaan ei kerro kehitystiimille, kuinka kehitysjonoa työtetään. Ammatillisista ja monitaitoisista koostuva kehitystiimi on kooltaan riittävän pieni, jotta se pysyy ketteränä, mutta myös riittävän suuri, jotta sprintissä saadaan kaikki merkittävä työ tehdyksi. Vaikka tiimit voivat koostua useista eri taidoista ja työn painopisteistä, niin vastuu kehityksestä kuuluu koko kehitystiimille. [Schwaber ja Sutherland, 2017, s. 7]

Scrummaster on vastuussa Scrum-menetelmän edistämisestä ja tukemisesta [Schwaber ja Sutherland, 2017, s. 7]. Scrummaster pitää huolen, että kaikki scrum-tiimin jäsenet käyttävät ja ymmärtävät Scrumia. Scrummaster on scrum-tiimin palveleva johtaja, joka auttaa kaikin mahdollisin keinoin scrum-tiimiä onnistumaan sekä suojelee, opettaa ja opastaa scrum-tiimiä. [Sutherland, 2010, s. 16]

Scrum-menetelmässä käytetään ennalta sovittuja tapahtumia, joilla luodaan säännöllisyyttä ja minimoidaan ylimääräiset palaverit. Nämä kaikki tapahtumat ovat ajallisesti rajattuja eli niillä on jokin maksimipituus. Scrum-menetelmän keskiössä on sprintti, joka on pisimmillään kuukauden pituinen tai lyhyempi ajanjakso, jonka aikana tuotetaan käyttökelpoinen ja potentiaalisesti julkaisukelpoinen inkrementti. Inkrementillä tarkoitetaan tuoteversiota ja se voi koostua useasta julkaisusta. Sprinteillä on sama pituus koko kehityksen ajan ja uusi sprintti alkaa aina välittömästi, kun edellinen sprintti on päättynyt. Sprintit koostuvat eri osista, joihin lukeutuu sprintin suunnittelupalaveri (eng. sprint planning), päivittäispalaverit (eng. daily scrum), kehitystyö (eng. development), sprintin katselmointi (eng. sprint review) ja sprintin retrospektiivi (eng. sprint retrospective). [Schwaber ja Sutherland, 2017, s. 9] Kuva 2 havainnollistaa Scrum-prosessin.



Kuva 2. Scrum-prosessi. Kuva muokattu lähteestä [Sutherland, 2010, s. 11].

Sprintissä tehtävä työ suunnitellaan aluksi sprintin suunnittelupalaverissa. Suunnitelma tehdään yhteistyössä koko Scrum-tiimin kanssa. Suunnitteluvaiheessa Scrum-tiimin tulee vastata kahteen oleelliseen kysymykseen, joista ensimmäinen on: Mitä sprintissä tehdään? Ensin kehitystiimi antaa ennusteen kaikesta toiminnallisuudesta, joka ehditään toteuttaa suunniteltavan sprintin aikana. Tuotteen omistaja puolestaan keskustelee kehitystiimin kanssa tavoitteista ja kehitettävän tuotteen kehitysjonon kohdista, jotka toteuttamalla päästään sprintin tavoitteisiin. Toinen kysymys johon Scrum-tiimin tulee vastata suunnittelussa, on: Miten valittu työ toteutetaan? On oleellista suunnitella, miten tehtävä työ toteutetaan valmiiksi inkrementiksi. Sprintin kehitysjoono muodostuu sprinttiin valituista kehitysjonon kohdista ja suunnitelmasta niiden toteuttamiseksi. [Schwaber ja Sutherland, 2017, s. 10]

Päiväpalaverit ovat tärkeässä roolissa Scrum-menetelmässä, koska näillä pyritään tekemään työstä läpinäkyvää. Kehitystiimi kokoontuu enintään 15 minuutin ajaksi ja tarkastelee edellisen päiväpalaverin jälkeen tehtyjä töitä ja suunnittelemaan seuraavat tehtävät

työt. [Schwaber ja Sutherland, 2017, s. 12] Jokainen tiimiläinen vastaa palaverissa myös kolmeen kysymykseen [Schwaber ja Sutherland, 2017, s. 12]:

1. Mitä tein eilen auttaakseni kehitystiimiä saavuttamaan sprintin tavoitteen?
2. Mitä aion tehdä tänään auttaakseni kehitystiimiä saavuttamaan sprintin tavoitteen?
3. Havaitsenko mitään estettä, joka estää minua tai kehitystiimiä saavuttamasta sprintin tavoitteen?

Sprintin loputtua valmistuneen inkrementin tulee olla ”valmis”-tilassa ja käyttökelpoisessa kunnossa. Tämä ”valmiin” määritelmä on Scrum-tiimin itse määrittelemä. Sprintin lopussa pidetään sprintin katselmointi, jossa tarkastelun keskiössä on kehitetty inkrementti. Lisäksi katselmoinnissa tehdään mahdolliset muutokset tuotteen kehitysjonoon. Scrum-timi ja mahdolliset sidosryhmät keskustelevat siitä, mitä sprintissä on saatu tehtyä. Katselmoinnin lopuksi päätetään, mitä seuraavaksi aletaan kehittämään arvon optimimiseksi. Sprintin katselmoinnin jälkeen ja ennen seuraavaa sprinttiä pidetään vielä sprintin retrospektiivi. Tässä retrospektiivissä Scrum-tiimi tarkastelee toimintaansa ja miettii mahdollisia parannuksia kehitysprosessiin, jotka puolestaan toteutetaan seuraavassa sprintissä. Sprintin retrospektiivin jälkeen aloitetaan uuden sprintin suunnittelu sprintin suunnittelupalaverilla. [Schwaber ja Sutherland, 2017, s. 18]

2.2.2 Lean

Lean-ajattelun juuret ovat 1940-luvun lopun Japanissa, jolloin autoja valmistava yritys Toyota kohtasi ongelmia tuotannossaan, koska ihmisillä ei ollut paljon rahaa ja autojen tuli olla edullisia. Massatuotanto oli halvin tapa tehdä autoja, mutta tämä tarkoitti tuhansien samojen autojen tekemistä ja Japanin markkinat eivät olleet tarpeeksi isot tälle tarpeelle. Ratkaisuksi ongelmaan keksittiin Toyota Production System, joka tarjosi uuden tavan ajatella tuotantoa, logistiikkaa ja lopulta tuotteen kehitystä. Uuden ajattelutavan isänä oli Taiichi Ohno. [Poppendieck ja Poppendieck, 2003] Ohnon uuden ajattelutavan keskiössä oli hukan poistaminen sekä arvon tuottaminen asiakkaalle [Poppendieck ja Poppendieck, 2003; Janes, 2015]. Ohno uskoi, että tuote tulisi tehdä heti kun asiakas on tehnyt tilauksen, eikä tehdä tuotteita ennakoon odottamaan tilauksia. Toimituksen tulee olla nopea tilauksen syntyessä, joten kaikki ylimääräinen hukka kehityksessä tulee hävittää. Hukalla tarkoitetaan ylimääräisiä toimintoja, jotka tuovat turhia kustannuksia ja hidastavat prosesseja. [Poppendieck ja Poppendieck, 2003]

Lean-ajattelu on käännetty myös ketterän ohjelmistokehityksen puolelle. Mary ja Tom Poppendieck ovat kääntäneet Ohnon kehittämän teollisen Lean-ajattelun ohjelmistokehityksen puolelle kirjoissaan, joissa he tuovat esille Lean-pohjaisen ketterän kehityksen

[Janes, 2015]. Seitsemäksi keskeiseksi Lean-ajattelun omaavalle ketterälle ohjelmistokehitykselle luetellaan seuraavat [Poppendieck ja Poppendieck, 2006]:

1. Hukan poistaminen
2. Jatkuva laadunvarmistus
3. Tietämyksen luominen
4. Päätöksenteon lykkääminen
5. Nopeus
6. Tiimityö
7. Kokonaisuuden optimointi

Lean-ajattelun keskiössä on kaiken hukan poistaminen. Lisäksi ketterä ohjelmistokehitys itsessään pyrkii etsimään ja poistamaan hukat. Tämän saavuttamiseksi on tärkeää tunnistaa, mistä hukka syntyy. Hyvä tapa aloittaa hukan etsiminen on kaikki se, mikä ohjelmistokehityksessä ei ole analysointia tai ohjelmointia. Tulee selvittää, mitkä prosessit eivät tuo arvoa asiakkaalle. Shigeo Shingo on yksi Toyota Production Systemin avainkehittäjistä ja hän on tunnistanut seitsemän hukkaa tuotannossa, jotka voidaan kääntää myös ohjelmistokehityksen puolelle. Taulukko 1 havainnollistaa alkuperäiset seitsemän hukkaa teollisessa tuotannossa sekä ohjelmistotuotannossa. [Poppendieck ja Poppendieck, 2003]

Taulukko 1. Seitsemän hukkaa Lean-ajattelussa
[Poppendieck ja Poppendieck, 2003].

Seitsemän hukkaa teollisessa tuotannossa	Seitsemän hukkaa ohjelmistokehityksessä
Varastot	Osittain tehty työ
Ylimääräinen prosessointi	Ylimääräinen prosessi
Ylituotanto	Ylimääräinen ominaisuus
Kuljetus	Työtehtävän vaihtaminen
Odottaminen	Odottaminen
Liike	Liike
Vika	Vika

Ohjelmistokehityksessä ensimmäisenä hukkana mainitaan osittain tehty työ. Osittain tehty työ voi olla esteenä jatkokehitykselle ja ei välttämättä toimi ollenkaan. Tämän kaltaisen työn on usein suuri taloudellinen riski. Osittaisen työn hävittäminen on hukan hävittämistä ja riskien vähentämistä. Ylimääräinen prosessin tuoma hukka näkyy etenkin turhassa paperityössä. Dokumentointi kehitettävien ominaisuuksien osalta tulisi tehdä vasta sille osoitettua iteraatiota ennen. Lisäksi ylimääräiset ominaisuudet nähdään

hukkana. Ylimääräisien ominaisuuksien lisääminen lisää muutakin ohjelmointitöitä ja ohjelmisto monimutkaistuu aivan turhaan. Työtehtävien vaihtaminen puolestaan luo hukkaa, koska tällöin työntekijä joutuu käyttämään aikaa uuden työtehtävän omaksumiseen. Työtehtävät tulee tehdä yksi kerrallaan hukan poistamiseksi. Yksi isoimmista hukista ohjelmistokehityksessä on asioiden odottaminen. Odottamista tapahtuu monessa asiassa, kuten projektin aloituksessa, arvioissa, testauksessa ja tuotannossa. Liike nähdään myös hukkana. Ihmisten sekä työkalujen tulee olla mahdollisimman nopeasti saatavilla, jotta turhalta liikehukalta vältytään. Viimeisenä hukkana ovat viat. Viat tulee löytää mahdollisimman nopeasti ohjelmistoista testauksilla, jotta niistä ei synny suurta hukkaa. [Poppendieck ja Poppendieck, 2003]

Hukan poistamisen lisäksi jatkuva laadunvarmistus kuuluu Lean-periaatteisiin. Virheitä tulee välttää koko ohjelmistokehityksen aikana rakentamalla laadukasta koodia, eikä vain turvautua myöhempään testaukseen. Kolmantena on tietämyksen luominen, jota kasvatetaan ohjelmiston kehittämisessä asteittain tarkentamalla sitä. Ohjelmiston vaatimuksia sekä koodia ei lyödä lukkoon ennen, kuin kehitystiimillä on tarvittava tietämys niistä. Lopullisten päätöksien tekemistä ohjelmistokehityksissä tulee lykätä mahdollisimman pitkälle. Päätökset tulee tehdä viime hetkellä, koska tällöin suurin osa tiedoista päätöksen tueksi on saatu kasaan. Tavoitteena tällä on välttää jo tehdyn työn peruuttamista. Nopeudella Lean-ajattelussa puolestaan tarkoitetaan ohjelmistoversioiden toimittamista asiakkaille niin nopeasti ja usein, etteivät he ehdi muuttaa mieltään. Tiimityötä korostetaan Lean-ajattelussa. Ohjelmistot kehitetään tiimeissä, jotka sisältävät tasokkaita johtajia. Yleensä onnistunut ohjelmisto on mahdollinen huippuluokan johdon ansiosta. Tiimin jäsenten tulee myös kunnioittaa toisiaan. Kunnioituksella tarkoitetaan kohtuullisten tavoitteiden antamista ja luottamista itseohjautuvuuteen tavoitteiden saavuttamiseksi. Viimeisenä Lean-periaatteena on kokonaisuuden optimointi. Ohjelmistokehityksen kaikki vaiheet ja näiden väliset siirtymät tulee tehostaa, eikä keskittyä vain yksittäisiin vaiheisiin. [Poppendieck ja Poppendieck, 2006]

Womack ja Jones [2003] ovat esittäneet viiden askeleen prosessin, kuinka Lean-menetelmä voidaan ottaa käyttöön (Kuva 3). Ensimmäisenä tulee tunnistaa tuotteen arvo asiakkaan näkökulmasta. Seuraavaksi prosessissa kartoitetaan koko arvoketju (eng. value stream) ja poistetaan ne vaiheet, jotka eivät luo arvoa. Kolmantena luodaan arvoa lisäävistä työvaiheista tiivis asiakasta kohti virtaava järjestelmä. Kun tämä virtaus on saatu aikaan, annetaan asiakkaan vetää arvoa järjestelmästä. Kun arvo on määritelty, arvoketjut kartoitettu, arvoa luomattomat vaiheet poistettu sekä vetoperiaate esitetty, aloitetaan prosessi uudestaan ja tätä jatketaan, kunnes täydellinen arvo saadaan luotua ilman hukkaa. [Womack ja Jones, 2003, 16-26]



Kuva 3. Lean-prosessi. Kuva muokattu lähteestä [Lean Enterprise Institute, 2019].

2.2.3 Kanban

Kanban on oleellinen menetelmä, kun halutaan toteuttaa Lean-ajattelua käytännössä. Kanban on yksi tärkeä johtamistyökalu Lean-tuotannossa. [Ikonen et al., 2011, s. 306] Kanban sana tulee japanin kielestä ja tarkoittaa taulua. Kanbanin juuret ovat 1940-luvun lopusta 1950-luvun alkuun, kun Taiichii Ohno kehitti Kanbanit kontrolloimaan prosessien välistä tuotantoa ja implementointia oikeaan aikaan (eng. Just in Time, JIT) Toyotan tuotannoissa. Kanbanin avulla Ohno pystyi minimoimaan prosessien työmäärät ja vähentämään varastokuluja. [Gross ja McInnis, 2003, s. 1-2] Kanban on ajoitusjärjestelmä, jonka avulla voidaan määrittää, mitä pitää tuottaa, milloin ja missä määrissä [Gross ja McInnis, 2003, s. 1-2; Ikonen et al., 2011, s. 306].

Kanban laittaa projektitiimit visualisoimaan työvirtaukset, vähentämään tehtäviä työmääriä eri vaiheissa ja mittaamaan työtehtäviin käytettävää aikaa [Ikonen et al., 2011, s. 306]. Ohjelmistokehityksessä Kanbanin käyttö perustuu työvirtauksien esittämiseen taulussa, joka sisältää sarakkeet eri työvaiheille (Kuva 4). Taululla käytetään Kanban-kortteja, jotka kuvaavat työtehtäviä ja nämä asetetaan taulun sarakkeisiin kuvaamaan työn tilaa. Kun työtehtävä saavuttaa vaatimuksensa, siirretään työtehtävän Kanban-kortti seuraavaan sarakkeeseen. Kanban-kortti liikkuu taululla vasemmalta oikealle, näin työvirta voidaan visualisoida. Tämä mahdollistaa koko projektin tilan sekä projektin kulun näkemisen helposti. [Poppendieck ja Cusumano, 2012, Ikonen et al., 2011, s. 306]

havaitaan ajoissa. Lisäksi helppokäyttöinen ohjelmisto vaatii vähemmän käyttäjäkoulutusta, vähemmän käyttötukea ja vähemmän ylläpitotyötä. [Bias ja Mayhew, 2005, s. 29-32]

Käyttäjakeskeinen suunnittelu laittaa käyttäjän kehitettävän tuotteen suunnitteluprosessin keskiöön. Suunnittelijat pyrkivät ymmärtämään mahdollisimman tarkasti käyttäjän tarpeet, toiveet ja rajoitteet koskien kehitettävää tuotetta, ja näiden pohjalta tekemään suunnitteluratkaisut, jotka kohtaavat käyttäjän tarpeet. [Pratt ja Nunes, 2012, s. 12] Käyttäjakeskeinen suunnittelu vaatii suunnittelijoita sekä analysoimaan että ennustamaan, miten käyttäjät käyttävät tuotetta, mutta myös testaamaan omia suunnitteluratkaisuja oikeassa käyttöympäristössä oikeilla käyttäjillä. Testaaminen on oleellinen komponentti käyttäjakeskeisessä suunnittelussa, koska usein on vaikeaa ymmärtää, kuinka käyttäjät hahmottavat, ymmärtävät ja käyttävät kehitettyjä suunnitteluratkaisuja. [Pratt, ja Nunes, 2012, s. 12]

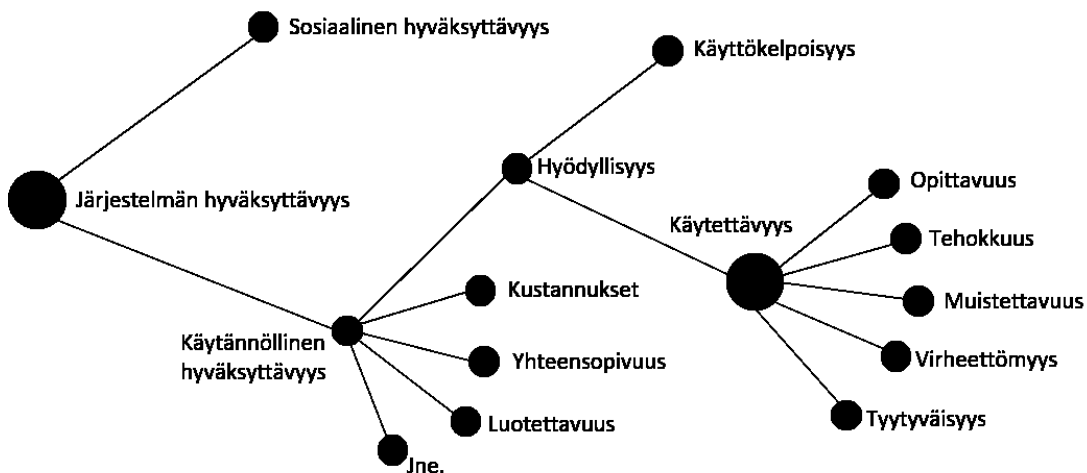
Käyttäjakeskeistä suunnittelua voidaan määritellä kansainvälisellä ISO 9241-210 standardilla [2010], jota tullaan käyttämään pitkälti tämän tutkimuksen teoreettisena viitekehystenä. Käyttäjakeskeinen suunnittelu (eng. user-centered design, UCD) on standardin mukaan suunnittelumenetelmä, jolla kehitetään vuorovaikutteisista ohjelmistoista käyttökelpoisempia, keskittymällä ohjelmiston käyttöön ja hyödyntämällä ihmisläheisiä tekijöitä. Apuna käytetään käytettävyyteen liittyvää tietoutta ja tekniikoita. Vuorovaikutteisella ohjelmistolla tarkoitetaan ohjelmistoa, jolle käyttäjä antaa syötteitä, ja joka antaa käyttäjälle palautetta. On huomioitava, että termin ”käyttäjakeskeinen suunnittelu” lisäksi käytetään termiä ”ihmisläheinen suunnittelu” (eng. human-centered design), koska se korostaa enemmän ihmistä. Termejä kumminkin käytetään toistensa synonyymeinä. [ISO 9241-210, 2010, s. 2] Tässä tutkimuksessa käytetään vain termiä ”käyttäjakeskeinen suunnittelu”.

Kuten edellä todettiin, käytettävyys kuuluu käyttäjakeskeisyyden peruskäsitteisiin. Käytettävyys tuotteen tai ohjelmiston ominaisuutena kuvaa, kuinka sujuvasti sen toimintoja käyttäjä käyttää päästäkseen haluttuun päämäärään. Käytettävyys voidaankin tiivistää ihmisen ja koneen vuorovaikutukseksi. Käytettävyys voi olla hyvää tai huonoa. Esimerkkinä oven käyttöliittymästä ei välttämättä saa selville, mistä reunasta se avataan, jolloin käytettävyyden voi todeta huonoksi. Pienillä muutoksilla voidaan parantaa käytettävyyttä hyväksi, kun oveen esimerkiksi lisätään visuaalinen vihje. [Kuutti, 2003, s. 13] ISO 9241-210 [2010, s. 3] määrittää käytettävyyden vaikutukseksi, tehokkuudeksi ja tyytyväisyydeksi tietyssä käyttöympäristössä, jossa tietyt käyttäjät saavuttavat tietyt tavoitteet. Tämänkaltainen standardi on ajanmukainen, koska käyttäjakeskeiset suunnittelumenetelmät

korostavat työtehtävien ja ympäristöjen merkitystä käytettävyyden suunnittelussa. [Kuutti, 2003, s. 15]

Käyttäjän kannalta parannetulla käytettävyydellä tarkoitetaan sitä, että opetteluun kuluu entistä vähemmän aikaa ja käyttäjä pääsee vielä nopeammin tekemään tehtäviään. Kun kyseessä on harvoin toistuvan tehtävän tekeminen, käyttäjä muistaa tämän tekemisen paremmin. Käyttäjä pystyy työskentelemään tehokkaammin, koska tietyssä ajassa saadaan valmiiksi enemmän tehtäviä kuin vanhalla tavalla tehtäessä. Virheiden määrä myös vähenee, kun käytettävyys on hyvää. Käyttäjä ei myöskään joudu takkuamaan käyttöliittymän hankaluuksien kanssa, vaan kokee viihtyvänsä tehtävissään hyvän käytettävyyden ansiosta. [Ovaska et al., 2005, s. 14]

Käytettävyyden Nielsen [1993, s. 25] kiteyttää siihen, miten hyvin käyttäjä voi käyttää ohjelmistoa. Käytettävyyteen yhdistetään ohjelmiston opittavuus, tehokkuus, muistettavuus, virheettömyys ja tyytyväisyys (Kuva 5).

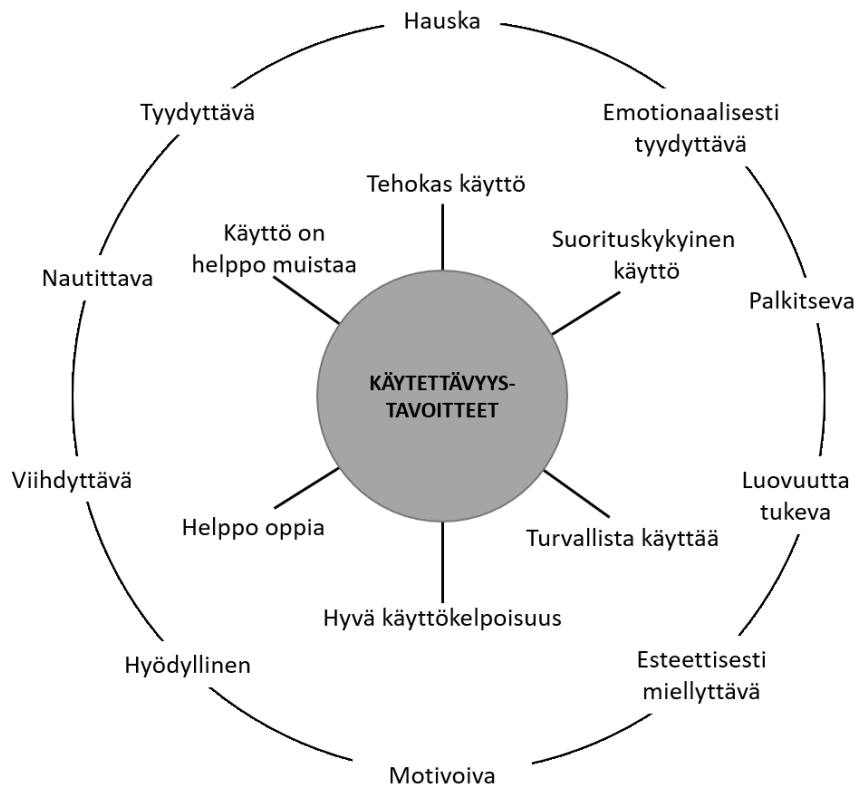


Kuva 5. Käytettävyyden käsitekartta. Kuva muokattu lähteestä [Nielsen 1993, s. 25].

Ohjelmiston tulisi olla helposti opittava, jotta käyttäjä voi nopeasti aloittaa työnteon ohjelmiston kanssa. Ohjelmiston käytön tulee olla tehokasta, jotta korkean tason tuottavuus voidaan saavuttaa. Muistettavuudella puolestaan tarkoitetaan sitä, että ohjelmiston käytön tulisi olla helposti muistettavissa pidempienkin käyttötaukojen jälkeen. Ohjelmiston ei pitäisi sisältää virheitä ja virheistä toipumisen tulisi olla helppoa. Tyytyväisyydellä tarkoitetaan sitä, että ohjelmistoa tulisi olla mukava käyttää. Nielsen erottaa vielä käytettävyyden ja käyttökelpoisuuden toisistaan käytettävyyden käsitekartassaan (Kuva 5). Käyttökelpoisuudella tarkoitetaan ohjelmiston mahdollisuuksia toimia tietyssä tehtävässä tavoitteen saavuttamiseksi. [Nielsen, 1993, s. 25-26] Käyttökelpoisuus selviää vasta itse käytännön tehtävissä, eikä sitä pysty havainnollistamaan käytettävyystesteillä, kuten käytettävyyttä. Käyttökelpoisuus ja käytettävyys kummatkin vaikuttavat ohjelmistosta

saatavaan hyötyyn. [Ovaska et al., 2005, s. 3] Kuvasta 5 selviää myös muut järjestelmän hyväksyttävyyteen vaikuttavat asiat, joita ovat sosiaalinen hyväksyttävyys ja käytännöllinen hyväksyttävyys.

Käyttäjäkokemus (eng. user experience, UX) on tärkeä käsite suhteessa käyttäjäkeskeiseen suunnitteluun. ISO 9241-210 [2010, s. 3] määrittää käyttäjäkokemuksen yksilön käsitteiksi ja reaktioiksi, jotka muodostuvat ohjelmiston käytöstä. Käyttäjä käyttää ohjelmistoa ja hänen tuntemuksia voidaan tarkkailla ja mitata. Käytettävyyteen verrattuna käyttäjäkokemus nähdään laajempänä kokonaisuutena, jossa käyttäjän toiminnat, tunteet ja oletukset ovat tarkkailussa, kun taas puolestaan käytettävyydessä tarkastellaan vain käyttäjän taitoja suorittaa määritelty tehtävä onnistuneesti. [Albert ja Tullis, 2013, s. 5] Käyttäjäkokemukseen voidaan lukea seuraavat tuntemukset: tyydyttävyys, hauskuus, emotionaalinen tyydyttävyys, palkitsevuus, luovuutta tukeva, esteettisesti miellyttävä, motivoiva, hyödyllisyys, viihdyttävyys ja nautittavuus. Näiden katsotaan olevan avainasemassa käyttäjäkokemuksen luomisessa ja ne eroavat käytettävyydestä tavoitteista. Kuva 6 havainnollistaa käytettävyyden ja käyttäjäkokemuksen tavoitteet. [Preece et al., 2002, s. 18-20]



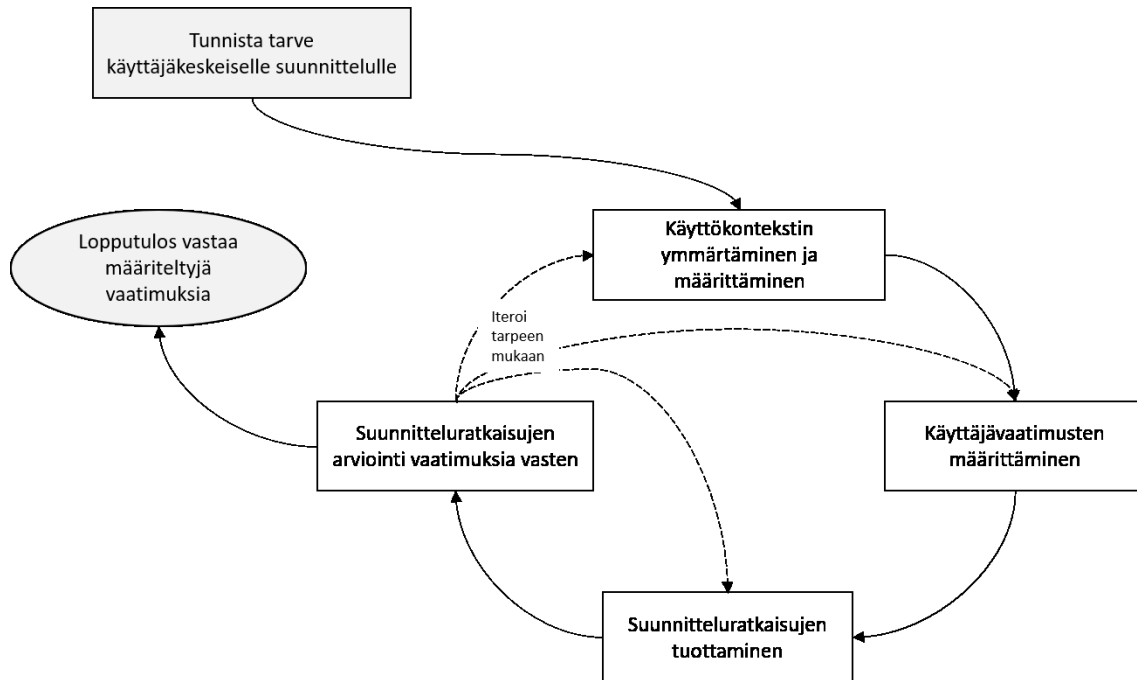
Kuva 6. Käytettävyys ja käyttäjäkokemus tavoitteet. Käytettävyydestä tavoitteet ovat keskellä ja käyttäjäkokemustavoitteet ovat ulkoringissä. Kuva muokattu lähteestä [Preece et al., 2002, s. 19].

Käyttäjäkeskeisen suunnittelun toteuttamisen avuksi on luotu erilaisia periaatteita ja tavoitteita. Gulliksen et al. [2003] tiivistävät onnistuneen käyttäjäkeskeisen suunnittelun 12 päätavoitteeseen [Gulliksen et al., 2003, s. 401-403]:

1. Ensin ja tärkeimpänä tulee fokus pitää käyttäjässä koko projektin ajan, jossa käyttäjän tarpeet ohjaavat kehitystä.
2. Käyttäjien tulee osallistua ajoissa projektiin ja olla mukana koko projektin ajan.
3. Järjestelmän kehityksen tulee olla koko ajan kehittyvää iteraatioiden kautta. Kehitys on evolutionääristä.
4. Suunnitelmat tulee esittää tarpeeksi yksinkertaisesti.
5. Prototyyppejä tulee tehdä mahdollisimman aikaisin ja paljon käyttäjien kanssa.
6. Käytön arviointi toiminnallisessa kontekstissa ja käyttöympäristössä.
7. Eksplisiittiset ja tietoiset suunnittelutoimet.
8. Ammattitaitoinen asenne suunnittelussa.
9. Käytettävyyssiantuntijoita tulisi käyttää ja pitää mukana projektin alusta loppuun asti.
10. Holistinen suunnittelu on oleellista eli kaikkia projektin osia kehitetään rinnakkaisesti. Järjestelmä on kokonaisuus, joka muodostuu monista eri osista, joten rinnakkaisuus on välttämätöntä.
11. Suunnittelumallia tulee muokata projektin tarpeisiin. Valmista muottia ei ole, joka sopii kaikille.
12. Lopuksi tärkeänä on käyttäjäkeskeisen asenteen luominen ja säilyttäminen. Kaikkien projektissa mukana olevien tulee olla tietoisia käytettävyydestä ja käyttäjäkeskeisyydestä, jotta niitä voidaan implementoida projektissa.

2.3.1 Prosessi

Suunnitteluprosessit ovat monipolvisia ja monitahoisia tapahtumasarjoja. Erilaisilla yrityksillä ja tuotteilla on omat suunnitteluprosessit, mutta niissä on myös samoja piirteitä ja säännönmukaisuuksia. [Hyysalo, 2009, s. 55] Käyttäjäkeskeisen suunnittelun prosessi on iteratiivinen (Kuva 7). Iteratiivisuudella tarkoitetaan tässä toistettavia jaksoja, kunnes haluttu lopputulos on saavutettu. Prosessi alkaa aina käyttäjäkeskeisen suunnitteluprosessin suunnittelulla, jonka jälkeen prosessi siirtyy iteratiiviseen suunnittelusykliin. Prosessi jatkuu käyttökontekstin ymmärtämisellä ja määrittämisellä. Tämän jälkeen käyttäjän vaatimukset määritellään, suunnitteluratkaisut toteutetaan ja lopuksi arvioidaan. Tätä iteratiivista prosessia jatketaan, kunnes suunnitteluratkaisut täyttävät käyttäjävaatimukset. [ISO 9241-210, 2010, s. 6-11]



Kuva 7. Käyttäjakeskeisen suunnittelun prosessi. Kuva muokattu lähteestä [ISO 9241-210, 2010, s. 11].

Prosessin tuotokset vaihtelevat sen vaiheiden mukaan. Käyttökontekstin ymmärtäminen ja määrittäminen tuottavat käyttökontekstin kuvauksen. Käyttäjävaatimusten määrittäminen puolestaan tuottaa käyttökontekstin vaatimukset, käyttäjän tarpeiden kuvauksen ja käyttäjävaatimusten kuvauksen. Suunnitteluratkaisujen toteuttaminen tuottaa käyttäjän toimien kuvauksen, käyttöliittymä vaatimukset ja implementoidun käyttöliittymän. Viimeisenä aktiviteettina on suunnitteluratkaisujen arviointi suhteessa vaatimuksiin, jonka tuotoksina syntyy arvioidut tulokset, testitulokset ja tulosten tarkkailu. [ISO 9241-210, 2010, s. 4-5] Taulukko 2 havainnollistaa käyttäjakeskeisen suunnittelun aktiviteetit ja tulokset.

Taulukko 2. Käyttäjakeskeisen suunnittelun aktiviteetit ja tuotokset [ISO 9241-210, 2010, s. 5].

Aktiviteetti	Tuotos
Käyttökontekstin ymmärtäminen ja määrittäminen	Käyttökontekstin kuvaus
Käyttäjävaatimusten määrittäminen	Käyttökonteksti vaatimukset Käyttäjän tarpeiden kuvaus Käyttäjävaatimusten kuvaus
Suunnitteluratkaisujen tuottaminen, jotka vastaavat vaatimuksia	Käyttäjän toimien kuvaus Käyttöliittymävaatimukset Implementoitu käyttöliittymä
Arvioi suunnitteluratkaisut suhteessa vaatimuksiin	Arvioidut tulokset Testitulokset Tulosten tarkkailu

2.3.2 Käyttäjäkeskeinen lähestymistapa

Käyttäjäkeskeisessä suunnittelussa tulee tunnistaa sen suuntaukset, jotka painottavat eri asioita käyttäjien huomioimiseksi eli kuinka käyttäjä on mukana suunnittelussa. Käyttäjien mukana oleminen voidaan jakaa neljään eri lähestymistapaan: käyttäjäkeskeiseen suunnitteluun (eng. user-centered design), osallistuvaan suunnitteluun (eng. participatory design), etnografiaan (eng. ethnography) ja kontekstuaaliseen suunnitteluun (eng. contextual design). [Kujala, 2003, s. 2-4]

Taulukko 3. Lähestymistavat käyttäjän osallistumiseen [Kujala, 2003, s. 3].

	Käyttäjäkeskeinen suunnittelu	Osallistuva suunnittelu	Etnografia	Kontekstuaalisen suunnittelu
Painopiste	Käytettävyys	Tasavertainen osallistuminen	Sosiaaliset näkökulmat työssä	Työn konteksti
Menetelmät	Tehtäväanalyysit, Prototyypit, Käytettävyysarvioinnit	Työpajat, Prototyypit	Tarkkailu, videoanalyysit	Tilannetutkimukset, prototyypit

Taulukosta 3 huomaa, että käyttäjäkeskeisen suunnittelun painopiste on käytettävydessä. Tavoitteena on luoda hyödyllinen ja käytettävä ohjelmisto. Pääperiaatteiksi luetellaan aikainen huomio käyttäjiin ja heidän toimiin, empiiriset mittaukset ja iteratiivinen suunnittelu. Lähestymistavassa suunnittelijat lähestyvät potentiaalisia käyttäjiä ja käyttäjät osallistuvat simulaatioihin ja prototyyppien käyttöön, joilla tehdään oikeita töitä. Käyttäjien suorituksia ja reaktioita tarkkaillaan, tallennetaan ja analysoidaan. Osallistuva suunnittelu puolestaan painottaa tasavertaista osallistumista. Käyttäjät osallistuvat analysoimalla vaatimuksia ja suunnittelemalla sosiaalisia sekä teknisiä rakenteita tukemaan yksilöiden ja organisaation tarpeita. Etnografinen menetelmä on sosiologinen lähestymistapa. Se on vaikutusvaltaisin tutkimusyhteisöissä. Etnografia kuvaa ihmisten aktiviteetteja ja kulttuuria keskittymällä ihmisten yhteistyön sosiaaliseen puoleen. Etnografia voidaan jakaa neljään periaatteeseen: se tapahtuu luonnollisessa ympäristössä, se perustuu holismiin, se kehittää kuvaavan ymmärryksen ja se on sidoksissa ihmisen näkökulmaan, jotka selvitetään havainnointien ja videoanalyysien avulla. Viimeinen lähestymistapa on kontekstuaalinen suunnittelu, jossa pääpaino on tutkia käyttäjiä heidän töissään. Käyttäjien työtehtäviä havainnoidaan ja jutellaan heidän työtehtävistään oikeassa työympäristössä. Ideana on tutkia työprosesseja sekä kuvata ja uudelleen suunnitella niitä. [Kujala, 2003, s. 3-4]

2.3.3 Toimintatavat ja välineet käyttäjäkeskeisessä suunnittelussa

Käyttäjäkeskeinen suunnittelu pitää sisällään useita erilaisia toimintatapoja ja välineitä, joilla sitä voidaan toteuttaa. Taulukkoihin 4, 5 ja 6 on kirjattu yleisimmät toimintatavat ja välineet käyttäjäkeskeisen suunnittelun eri vaiheissa. Vaiheina ovat tutkimusvaihe, suunnitteluvaihe ja arviointivaihe. Taulukoiden sisältö on koottu useasta eri lähteestä [Heinilä et al., 2005; Hyysalo, 2009; Kuutti, 2003; Nielsen, 1993; Ovaska et al., 2005; Pratt and Nunes, 2012; Rekha Devi et al., 2012] koskien käyttäjäkeskeisen suunnittelun toimintatapoja ja välineitä. Vaikka toimintatavat ja välineet on jaettu käyttäjäkeskeisen suunnittelun eri vaiheiden mukaan, niitä voi esiintyä myös muissa vaiheissa, ne vain painottuvat eritavalla eri vaiheissa enemmän.

Ensimmäisessä taulukossa (Taulukko 4), on kuvattuna käytetyt toimintatavat sekä välineet käyttäjäkeskeisen suunnittelun tutkimusvaiheessa. Tutkimusvaiheessa käytetään tilannetutkimusta (eng. contextual inquiry) kyselyitä (eng. survey), haastatteluja (eng. interview), havainnointia (eng. observation), työpajoja (eng. workshop), fokusryhmiä (eng. focus group), persoonia (eng. persona), skenaarioita (eng. scenario) ja tehtäväanalyysijä (eng. task analysis). Suurin osa toimintatavoista ja välineistä painottuu alun tutkimusvaiheeseen.

Taulukko 4. Käyttäjäkeskeisen suunnittelun tutkimusvaiheen toimintatavat ja välineet. Koottu lähteistä [Heinilä et al., 2005; Hyysalo, 2009; Kuutti, 2003; Nielsen, 1993; Ovaska et al., 2005; Pratt and Nunes, 2012; Rekha Devi et al., 2012].

Toimintatapa / Väline	Kuvaus
Tutkimusvaihe	
Tilannetutkimus	Tutkimusmenetelmä, jolla kerätään tietoa käyttäjistä heidän todellisessa työympäristössään. Tähän menetelmään kuuluu käyttäjän kontekstin selvittäminen. Menetelmässä yhdistyy käyttöympäristössä tapahtuvat haastattelut ja osallistuvat havainnoinnit.
Kysely	Kyselyt ovat yksi haastattelumuoto, jossa haastattelu on puettu kirjoitettuun muotoon, joka voidaan esimerkiksi postittaa tai laittaa verkkoon. Kyselyt ovat yleensä strukturoituja, jotta lomaketta on nopea täyttää ja tämän jälkeen analysoida. Lomakekysely mahdollistaa laajan osallistujajoukon käyttämisen. Voidaan saada määrällistä tietoa jo olemassa olevasta tehtävästä tai ohjelmistosta.
Haastattelu	Haastattelut ovat oleellinen tapa kerätä käyttäjätietoa, koska käyttäjien tekemisiä on vaikea saada selville ilman, että he itse kertovat omia tulkintojaan. Haastattelut ovat helposti toteutettavissa ja mukautettavissa. Ongelmiksi voi kuitenkin muodostua johdattelevat kysymykset ja erilaiset vääristymät. Haastattelumenetelmiä on myös erilaisia. Strukturoitu haastattelu on kysely muutettuna haastatteluksi. Teemahaastattelu taas on vapaampi, jossa kysymysrunkoa käydään läpi haastateltavan vastauksiin mukautuen, tehden tarkentavia kysymyksiä. Lisäksi haastattelut voivat olla puhelinhaastatteluja, ryhmähaastatteluja tai ryhmäkeskusteluja.
Havainnointi	Havainnointi tarkoittaa käyttäjätietojen keräämisessä ihmisten toimien tarkkailua heidän ympäristöissään. Näin tuotekehittäjä saa omakohtaisen yleistuntuman käyttäjistä, heidän käyttöympäristöistä ja toimista. Oleellisessa osassa havainnoinnissa on havaintojen kirjaaminen. Havainnoinnin vaaroihin lukeutuu liian nopea yleistäminen ja virhetulkinnat. Tärkeimpiin havainnointi menetelmiin lukeutuu passiivinen havainnointi, jossa seurataan tapahtumia ja tehdään muistiinpanoja. Havainnointihaastatteluissa havainnoija seuraa henkilöä ja kyselee samalla kysymyksiä. Osallistuvassa havainnoinnissa puolestaan havainnointia tehdään samalla, kun osallistutaan käyttäjän toimintaan. Käyttäjien toimista voidaan tehdä kuvauksia kuvaavien mallien avulla.
Työpaja ja fokusryhmä	Käyttäjäkunnan edustajia voi olla mukana arviointi- ja suunnittelutöissä. Näistä saadaan monet eri näkökulmat esille niin puolesta, että vastaan. Samalla saadaan kontakti käyttäjiin. Liian dominoivat persoonat voivat kuitenkin vaikeuttaa tilanteen etenemistä. Työpajat kuitenkin auttavat ymmärtämään, kuinka ryhmä käyttäjiä suhtautuu ohjelmistoon tai työympäristöön.
Persoonat	Persoonat ovat kuvauksia ohjelmiston käyttäjistä. Ne yleensä sisältävät kuvan, nimen ja erinäisiä kuvauksia käyttäjästä. Niissä tuodaan esille käyttäjän tavoitteet ja tarpeet, mitä käyttäjä haluaa ja mitkä asiat ärsyttävät häntä. Persoonien avulla muistutetaan kehitystiimejä, mitä ollaan suunnittelemassa ja kenelle.
Skenaario	Skenaariot ovat minimalistisia prototyyppejä, koska ne kuvaavat yhden interaktion. Skenaariokuvaus sisältää käyttäjän, ohjelmiston käytön, käytön lopputuloksen, erikoistilanteet sekä suoritusajan. Skenaarioilla pyritään selvittämään, miten käyttäjä käyttää tulevaa ohjelmistoa sekä saamaan käyttäjäpalautetta ennen prototyyppien rakentamista. Skenaariokuvaukset ovat hyviä alkuvaiheen suunnittelussa, koska niitä voidaan luoda ja muokata ennen käyttöliittymän suunnittelua. Skenaariokuvauksia voidaan käyttää myös käyttäjätestauksissa.
Tehtävänälyysi	Tehtävänälyysi on oleellinen suunnittelun alkuvaiheissa. Käyttäjien tavoitteet tulisi olla tutkittuna sekä kuinka käyttäjät suorittavat tehtävänsä nyt. Tehtävänälyysillä tunnistetaan, mitä ohjelmisto tarvitsee. Tehtävänälyysin tyypillinen lopputulos on listaus asioista, mitä käyttäjä haluaa saavuttaa ohjelmistolla tiedot, mitä tarvitaan tavoitteiden saavuttamiseen sekä vaiheet, mitä tulee suorittaa.

Seuraavassa taulukossa (Taulukko 5) kuvataan suunnitteluvaiheen toimintatavat ja

välineet. Näihin lukeutuu korttien lajittelu (eng. card sorting), piirrokset (eng. sketch), rautalangat (eng. wireframe) sekä prototyypit (eng. prototype).

Taulukko 5. Käyttäjäkeskeisen suunnittelun suunnitteluvaiheen toimintatavat ja välineet. Koottu lähteistä [Heinilä et al., 2005; Hyysalo, 2009; Kuutti, 2003; Nielsen, 1993; Ovaska et al., 2005; Pratt and Nunes, 2012; Rekha Devi et al., 2012].

Toimintatapa / Väline	Kuvaus
Suunnitteluvaihe	
Korttien lajittelu	Korttien lajittelua käytetään käyttäjän mielen maailman ymmärtämiseksi. Käyttäjää pyydetään asettamaan kortit, jotka vastaavat toimintoja kategorioiden alle. Saman kategorian alla olevat kortit tulisi todennäköisesti sisällyttää samaan ryhmään. Korttien lajittelu auttaa käyttöliittymän sisällön rakenteen suunnittelussa.
Piirros	Suunnitteluun on monia erilaisia lähestymistapoja. Piirrosten tekeminen on hyvä tapa tutkia erilaisia suunnittelulähestymistapoja. Ideat piirretään kynän ja paperin avulla tai taululle. Piirroksien avulla voidaan kommunikoida tiimin ja asiakkaan kanssa suunnitteluideoista. Piirrokset mahdollistavat monien eri vaihtoehtojen tutkimisen ja vastaavat moniin kysymyksiin, ennen varsinaista käyttöliittymäsuunnittelua.
Rautalanka	Rautalangat ovat ohjelmistojen pohjapiirroksia. Ne kuvaavat käyttöliittymän rakenteen, sisällön sijainnin ja käyttöliittymä elementit, kuten navigointi ja painikkeet, sekä kuinka nämä toimivat yhdessä. Rautalangat eivät yleensä sisällä lopullisia visuaalisia elementtejä. Rautalangat auttavat suunnitelmien kommunikoinnissa ja antavat suunnan suunnittelulle.
Prototyyppi	<p>Prototyyppiä voi olla monen eri tasoisia, eivätkä ne rajoitu yhteen tyyppiin. Prototyypeillä on mahdollista testata esimerkiksi uutta vuorovaikutuskonseptia tai kokonaista sovellusta. Erityisen tärkeänä pidetään uusien konseptien prototypointia varhaisessa vaiheessa suunnittelussa. Tämänkaltaiset prototyypit voivat olla todella yksinkertaisia ja nopeasti tehtyjä, joilla pyritään määrittämään esimerkiksi vuorovaikutuskonseptia. Prototyypit voivat helpottaa kommunikointia kehitystiimin kesken. Prototyyppien avulla voidaan varmistaa, että kaikki ovat ymmärtäneet suunnitteluratkaisut halutuilla tavoilla. Prototyypit konkretisoivat suunnitteluideat muotoon, jossa niitä on mahdollista testata ja kommentoida. Tällä tavalla voidaan jo varhaisessa vaiheessa löytää tuotteen ongelmia ja puutteita. Prototyyppien käytön takana on ajan ja rahan säästäminen sekä prototyyppien käyttäminen osana käyttäjätestauksia. Ongelmaksi voi kuitenkin muodostua väärin prototyyppien testaaminen tai nopeasti tehdyn prototyypin lopullisen tuotteen kehittämisessä, vaikka sen toteutus olisi ollut liian huolimaton.</p> <p>Prototyypit voidaan jakaa toiminnallisiin (interaktiivinen) prototyypeihin ja paperiprototyypeihin. Toiminnallinen prototyyppi on kaikista pisimmälle viety prototyyppi. Tämä on periaatteessa toimiva sovellus. Prototyyppien tekemiseen ei kuitenkaan tule käyttää liikaa aikaa, joten kustannuksia säästämiseksi on olemassa kolme tapaa: toiminnallisuuden karsiminen, ominaisuuksien karsiminen ja ohjelmistotyössä säätäminen. Aina ei kuitenkaan tarvita toiminnallista prototyyppiä, vaan ominaisuuksia ja käyttöliittymää voidaan testata paperiprototyypeillä. Tässä käytetään paperista käyttöliittymän kuvaa, jota voidaan testata sovelluksen loppukäyttäjillä. Testin järjestäjät esittävät käyttäjälle näytön kuvaa ja käyttäjä kertoo, mitä näkee. Näyttöön voidaan tehdä muutoksia lapuilla tai piirtämällä ja koko näyttö voidaan myös vaihtaa tarvittaessa. Lisäksi paperiprototyyppiä voidaan tehdä suunnittelutyökaluilla. Muihin prototyypeihin verrattuna paperiprototyypit ovat erityisen helppo, nopea ja halpa toteuttaa.</p>

Viimeisessä käyttäjäkeskeisen suunnittelun toimintatapojen ja välineiden taulukossa (Taulukko 6) kuvataan suunnitelmien arviointivaiheen toimintatavat ja välineet. Näihin

lukeutuu käytettävyysestaus (eng. usability testing) ja asiantuntija-arviot (eng. expert evaluation).

Taulukko 6. Käyttäjäkeskeisen suunnittelun arviointivaiheen toimintatavat ja välineet. Koottu lähteistä [Heinilä et al., 2005; Hyysalo, 2009; Kuutti, 2003; Nielsen, 1993; Ovaska et al., 2005; Pratt and Nunes, 2012; Rekha Devi et al., 2012].

Toimintatapa / Väline	Kuvaus
Arviointivaihe	
Käytettävyysestaus	<p>Käyttäjättestaukset oikeilla käyttäjillä on tärkeä käytettävyyssuunnittelu, eikä sitä voi juurikaan korvata, koska sillä saavutetaan tietoa siitä, miten käyttäjä käyttää ohjelmistoa ja mitä ongelmia ohjelmiston käytössä ilmenee. Käytettävyysestauksissa ohjelmiston kohderyhmää edustava testihenkilö suorittaa ohjelmistolla tai sen prototyyppillä etukäteen määritellyjä tehtäviä. Tämän pohjalta käytettävyysestauksien järjestäjät tekevät havaintoja käytettävyyssuunnittelusta ja käyttöliittymästä.</p> <p>Testauksien avulla pyritään siis saamaan esiin tuotteen mahdolliset muutostarpeet. Testauksien avulla pystytään paljastamaan jo varhaisissa vaiheissa tuoteidean puutteita ja ongelmia. Tulokset ovat selkeästi esitettävissä ja niistä päästään helposti parannustoimiin.</p>
Asiantuntija-arvio	<p>Asiantuntija-arvioinnissa asiantuntijaryhmä tai asiantuntija suorittaa käytettävyyden arvioinnin. Nämä arvoinnit voivat perustua erilaisiin heuristiikkalistoisiin, joiden avulla asiantuntija voi arvioida käyttöliittymää.</p> <p>Nielsenin heuristinen arviointi on yleisin asiantuntija-arviointimenetelmä. Menetelmän avulla ohjelmistosta etsitään käytettävyyssuunnittelun ongelmia. Pieni joukko arvioijia tutkii vuorotellen ohjelmistoa käytettävyyssuunnittelun eli heuristiikkojen avulla. Heuristiikkoja on kymmenen: palvelun tilan näkyvyys, palvelun ja tosielämän vastaavuus, käyttäjän kontrolli ja vapaus, yhtenäisyys ja standardit, virheiden estäminen, tunnistaminen ennen muistamista, käytön joustavuus ja tehokkuus, esteettinen ja minimalistinen suunnittelu, virheiden tunnistaminen sekä opastus ja ohjeistus.</p>

3. Suunnittelun ja ketteryyden integroinnin kirjallisuuskatsaus

Kaikkea mahdollista ei voi tutkimuksessa havainnoida niin, että tarkasteltava ilmiö tulisi kaikilta puolilta tutkituksi [Hirsjärvi et al., 2007, s. 85]. Parhaimpaan tapaan kuuluukin tapa rajata aihetta tutustumalla aiempiin aineistoihin. Kun selvittää, miten aihetta on aikaisemmin käsitelty, on mahdollista suunnata tutkimusta. Kirjallisuuskatsauksen avulla voidaan luoda tutkimukselle perusta. [Hirsjärvi et al., 2007, 85, s. 87] Tässä tapauksessa kyseessä on käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integrointi.

Tämän kirjallisuuskatsauksen tarkoitus on luoda kokonaiskuva aiemmista tutkimuksista koskien aihepiiriä. Lisäksi kirjallisuuskatsauksen pohjalta luodaan tämän tutkimuksen empiirisen tutkimuksen pohja. Katsauksen tuloksia käytetään apuna muodostamaan tutkimuksen haastattelumenetelmiä ja erityisesti haastattelujen aiheita. Lisäksi kirjallisuuskatsauksen tuloksia tullaan vertaamaan tämän tutkimuksen haastattelutuloksiin.

3.1 Tietokantahaut

On olemassa eri tyyppisiä tietokantoja. Tietokannat voivat olla yleisiä, tieteenalakohtaisia ja monen alan kattavia, joissa voi tehdä yleensä monimutkaisiakin hakuja. Ne voivat sisältää kokoelmatekstejä, tilastoja tai kokoelma viitteitä. Eri tieteenalojen tietokannat, joihin ei välttämättä ole vapaata pääsyä verkossa, mahdollistavat luotettavan tiedon löytymisen. [Hirsjärvi et al., 2007, s. 89-96] Kirjallisuuskatsauksen tietokannoiksi valittiinkin Tampereen yliopiston tarjoamat tietokannat: ACM, IEEE, ScienceDirect ja Springer. Kyseiset tietokannat valittiin, koska ne ovat luotettavia yliopiston tarjoamia tietokantoja, jotka ovat laajoja ja tarjoavat hyvät hakumahdollisuudet. Ennen tietokantojen varsinaista valintaan niillä toteutettiin alustavia hakukyselyitä. Nämä alustavat kyselyt antoivat tarpeeksi relevantteja hakutuloksia, joten kyseiset tietokannat voitiin valita kirjallisuuskatsauksen lähteiksi.

Tietokantahaut tehtiin aikavälillä 1.10.2018 – 2.12.2018. Aineiston hakukriteereiksi asetettiin seuraavat:

- Aineisto haetaan ajalta 2005-2018.
- Aineiston tuli olla artikkelimuodossa oleva tutkimus.
- Aineiston tuli käsitellä käyttäjäkeskeisyyden ja ketterän kehityksen integrointia.

Hakuprosessi aloitettiin luomalla mahdollisimman kattava hakusanayhdistelmä. Aluksi kokeiltiin monia eri tyyppisiä hakusanayhdistelmiä liittyen käyttäjäkeskeiseen suunnitteluun ja ketterään kehitykseen. Eniten suotuisia hakuja saatiin yhdistämällä sanat ”agile” sekä ”user-centered design”. Integraatio sana tiputettiin hakusanayhdistelmästä

kokonaan, koska se rajasi liikaa otollisia tuloksia. Parhaat hakutulokset tuottavat hakusanat ja valitut hakusanayhdistelmät olivat seuraavat:

- **Hakusanat, jotka tuottivat parhaat tulokset:** User-Centered Design, UCD
User experience, Agile, Scrum
- **Valitut hakusanayhdistelmät:** (User-Centered Design OR UCD OR User experience) AND (Agile OR Scrum), User-Centered Design AND Agile

Valituilla hakusanayhdistelmillä tehtiin tietokantahaut valittuihin tietokantoihin. Taulukossa 7 on kuvattuna tietokantahaut.

Taulukko 7. Tietokantahaut kirjallisuuskatsausta varten.

Tietokantahaut: Käyttäjäkeskeinen suunnittelu ja ketterä kehitys				
Tietokanta	Hakusanat	Muut hakuehdot	Hakutulos	Valitut
ACM Digital Library	(User-Centered Design OR UCD OR User experience) AND (Agile OR Scrum)	Väliltä 2005-2018	943	2
IEEE/IET Electronic Library (IEL) / IEL	(User-Centered Design OR UCD OR User experience) AND (Agile OR Scrum)	Väliltä 2005-2018	228	6
ScienceDirect	(User-Centered Design OR UCD OR User experience) AND (Agile OR Scrum)	Väliltä 2005-2018, tietojenkäsittely	303	1
SpringerLink	(User-Centered Design OR UCD OR User experience) AND (Agile OR Scrum)	Väliltä 2005-2018, tietojenkäsittely	572	2
Hakutulokset yhteensä ja valitut:			2046	11

Hakujen tuloksena oli yhteensä 2046 artikkelia, joista mukaan valittiin 11. Ensimmäinen artikkelien karsinta tehtiin otsikon, tiivistelmän ja avainsanojen perusteella. Tutkimuskysymyksessä on ketterän kehityksen ja käyttäjäkeskeisen suunnittelun integroiminen, joten artikkeleiden tuli käsitellä tätä. Artikkelit otettiin tarkempaan arvioon, jos siinä mainittiin myös suunnittelijan näkökulma aiheeseen. Katsauksessa löydettyt artikkelit painottivat kuitenkin käyttäjäkeskeistä suunnittelua ja ketterää kehitystä yleisellä tasolla, tutkijoiden omien havaintojen kautta ja yhdistetyistä eri näkökulmista. Näiden tietojen pohjalta relevanteiksi osoittautui 11 artikkelia (Liite 1).

Lopuksi artikkelien laatu vielä arvioitiin tarkastuslistalla, jossa kysyttiin kysymyksiä, kuten: Ovatko tutkimustulokset uskottavia? Lisääkö tutkimus ymmärrystä tutkittavana olevasta aiheesta? Vastaako tutkimus omiin tutkimuskysymyksiinsä? Laaduntarkistuslistassa oli yhteensä kymmenen kysymystä, joista jokaisesta sai maksimissaan yhden

pisteen. Maksimipistemäärä oli 10, joka käytännössä tarkoitti laadukkainta. Tutkimukseen ei otettu mukaan artikkeleita, jotka saivat laadunarvioinnissa alle 8 pistettä. Laadun tarkistuslista luotiin käyttäen apuna Spencerin et al. [2003] luomaa laadunarviointiviitekehystä. Laaduntarkistuslista on omana liitteenä (Liite 2). Laaduntarkastus osoittautui oleelliseksi, koska muun muassa Brhel et al. [2015, s. 164] selvittivät, että käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integrointia käsittelevät kirjallisuuskatsaukset olivat usein puutteellisia, koska valituille tutkimuksille ei tehty kunnollista laadunarviointia.

Eroavaisuudet aineistossa muodostuvat eri lähestymistavoista aiheeseen. Artikkeleista huomasi niiden ajanjaksoille tyypilliset teemat. Aluksi keskityttiin selittämään mallien prosesseja auki, jonka jälkeen kiinnostus nousi mallien yhdistämiseen ja tämän jälkeen niiden tulevaisuuden näkymiin. Monet artikkelit nostivat esiin ongelmia eikä vastauksia. Lisäksi onnistuneiden mallien yhdistämisestä ei ollut kattavasti tutkimusta. Kirjallisuuskatsauksia käyttäjäkeskeisen suunnittelun ja ketterän kehityksen yhdistämisestä on tehty paljon, mutta ne pääasiassa keskittyvät käytettäviin ohjelmistokehityskäytäntöihin ja harvoin kuvaavat integroitua prosessia ja ihmisiä, joihin prosessit vaikuttavat [Brhel et al., 2015, s. 164].

Aiheessa riittää selvästi tutkimista. Käyttäjäkeskeisyys ja ketteryys onnistuneesti yhdistettynä on yrityksille hyödyllinen kehitysmalli tehdä ohjelmistokehitystä, jonka takia sitä on syytä tutkia lisää ja uudesta näkökulmasta. Kirjallisuuskatsaus alkaa ketterän kehityksen ja käyttäjäkeskeisen suunnittelun integroinnin yleisellä tutkimisella. Tämän jälkeen luodaan katsaus käytänteisiin, joiden katsotaan kuuluvan ketterän kehityksen ja käyttäjäkeskeisen suunnittelun yhdistämiseen.

3.2 Ketteryyden ja käyttäjäkeskeisen suunnittelun yhdistäminen

Ketterä ohjelmistokehitys ja käyttäjäkeskeinen suunnittelu ovat molemmat tärkeitä kehitysprosesseja. Kuitenkin näiden kahden eri kehitysprosessin integroiminen keskenään yhdeksi prosessiksi on hankalaa tiettyjen eroavaisuuksien takia. Ketterä kehitys ja käyttäjäkeskeinen suunnittelu ovat keskenään hieman erilaisia prosesseja, mutta ne sisältävät yhteisiä kehitystavoitteita, mitkä toimivat hyvinä aloituskohtina näiden kahden prosessin yhdistämiseen. [Jurca et al., 2014, s. 24] Käyttäjäkeskeisen suunnittelun ja ketterän kehityksen keskeiset samankaltaisuudet ovat [Chamberlain et al., 2006, s. 144-145]:

- Kummatkin ovat iteratiivisia prosesseja.
- Ketterät menetelmät painottavat käyttäjää, rohkaisten osallistumista koko kehitysprosessin ajan. Käyttäjäkeskeisessä suunnittelussa keskitytään käyttäjään alusta alkaen.

- Kummatkin menetelmät korostavat tiimin yhtenäisyyttä.

Samankaltaisuuksien lisäksi menetelmillä on selviä eroja. Käyttäjäkeskeisen suunnittelun ja ketterän kehityksen keskeiset eroavaisuudet ovat [Chamberlain et al., 2006, s. 144-145]:

- Käyttäjäkeskeinen suunnittelu kannattaa tiettyjen suunnitteluvälineiden käyttämistä kommunikoinnin tukena kehittäjien kanssa. Ketterissä menetelmissä puolestaan pyritään vähäiseen määrään dokumentointia.
- Käyttäjäkeskeinen suunnittelu pyrkii ymmärtämään käyttäjiä kunnolla ennen kehitystyön alkamista. Ketterissä menetelmissä puolestaan kartetaan ennakoon tutkimista, koska tämä aika on pois ohjelmoinnilta.

Kuitenkin tarve toimittaa ohjelmisto asiakkaalle nopeasti jatkuvasti muuttuvassa liiketoimintaympäristössä ottaen samalla huomioon loppukäyttäjien tarpeet, saa käyttäjäkeskeisen suunnittelun ja ketterän ohjelmistokehityksen näyttämään lupaavalta yhdistelmältä, joka onkin saanut paljon huomiota viime vuosina [Brhel et al., 2015, s. 164]. Käyttäjäkeskeisyyden yhdistämistä ketterään kehitykseen on myös pohdittu, koska ketteryys pelkästään ei aina takaa ohjelmistolle hyvää käyttöliittymää [Fox et al., 2008, s. 63]. Kummatkin mallit voivat hyödyntää ja parantaa toisiaan. Käyttäjäkeskeiset menetelmät voivat parantaa ketteryyttä systemaattisilla menetelmillä, joilla tarkastellaan loppukäyttäjän tarpeita. Ketteryys voi puolestaan parantaa käyttäjäkeskeisyyttä useimmilla iteraatioilla, jotka puolestaan johtavat useampaan käytettävyyssarviointiin ja aikaiset palautteet ohjelmistosta voidaan implementoida ohjelmistoon yhä nopeammin. [Jurca et al., 2014, s. 25] Mallit nähdään yhteensopivina ja ne voivat toimia yhdessä, mutta ongelmiakin voi esiintyä, jos yhdistämisen periaatteet eivät ole kunnossa [Chamberlain et al., 2006, s. 152].

3.3 Käytänteet ja menetelmät mallien yhdistämiseksi

Kirjallisuuskatsauksessa mukana olleista tutkimuksista löytyi toistuvia aiheita integraation mahdollistamiseksi. Seuraavat aiheet havaittiin tutkimuksista:

- Integroitu prosessi
- Etukäteen suunnittelu
 - Esivaihesuunnittelu
 - Etukäteissuunnittelu
- Käyttäjäkeskeinen suunnittelija osana ketterää kehitystiimiä
 - Suunnittelijan roolit
 - Kommunikointi suunnittelijan ja kehittäjän välillä
- Välineet ja toimintatavat suunnittelussa

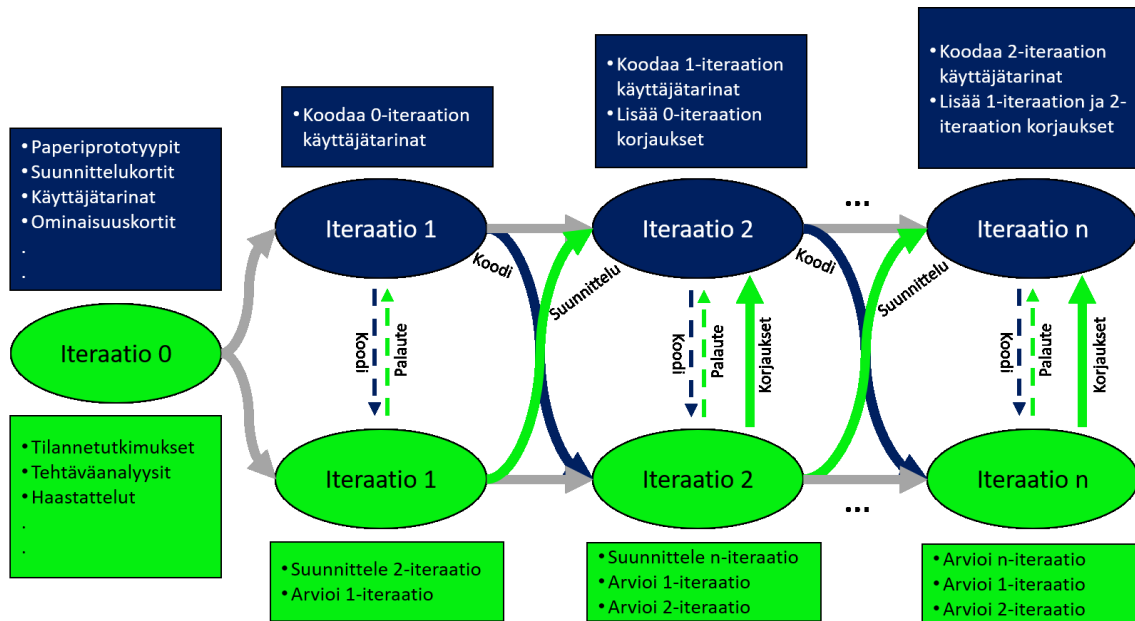
- Sidosryhmien mukana pitäminen
 - Loppukäyttäjä
 - Asiakas

Seuraavaksi luodaan katsaus näihin kirjallisuuskatsauksen aiheisiin. Aiheita aloitetaan käsittelemään aluksi korkealta tasolta integroidusta prosessista, jonka jälkeen siirrytään muihin käytänteisiin.

3.3.1 Integroitu prosessi

Ketteristä menetelmistä käytetyimmäksi kirjallisuuskatsauksessa selvisi Scrum-menetelmä [Anwar et al., 2014; Brhel et al., 2015; Chamberlain et al., 2006; Najafi ja Toyoshiba, 2008; Salah et al., 2014; Silva da Silva et al., 2013]. Integraatiossa mukana pidetään käytettävyys ja Scrum-periaatteet. Erityisesti käytettävyys periaatteet ja käytännöt huomioidaan, jotta ohjelmiston käytettävyys paranee ja samalla pyritään vähentämään ketterien aktiviteettien vaikutusta kehitykseen. [Anwar et al., 2014, s. 163]

Silva da Silvan et al. [2011] tekemän tutkimuksen pohjalta syntyi integroidun ketterän kehityksen ja käyttäjäkeskeisen suunnittelun viitekehys (Kuva 8). Tutkimuksessa havaittiin toistuvat teemat ja mallit sekä useimmin käytetyt toimintatavat ja välineet yhdistettyssä ketterässä ja käyttäjäkeskeisessä suunnittelussa. Tutkimus perustui laajaan kirjallisuuskatsaukseen, jossa tutkittiin 309 paperia, joista valittiin tutkimukseen 58 tarkempaan analyysiin. Suuri karsinta johtui siitä, että kaikki paperit eivät käsitelleet itse integraatiota. [Silva da Silva et al., 2011, s. 79] Silva da Silvan et al. [2011] luoma malli on hyvin samankaltainen Sy [2007], Najafi ja Toyoshiba [2008] sekä Fox et al. [2008] kehittämien mallien kanssa, jotka myös tutkivat käyttäjäkeskeisen suunnittelun ja ketterän kehityksen yhdistämistä.



Kuva 8. Integroidun ketterän kehityksen ja käyttäjäkeskeisen suunnittelun viitekehys.
Kuva muokattu lähteestä [Silva da Silva et al., 2011, s. 84].

Kuvassa 8 havainnollistetaan ketterän kehityksen ja käyttäjäkeskeisen suunnittelun integrointi. Suunnittelu- ja kehityksaktiviteetit osana ohjelmistokehitystä tulisi nähdä rinnakkaisesti yhdessä [Brhel et al., 2015, s. 172]. Ylempänä kulkee kehityspolku ja alimpana on suunnittelupolku. Nämä vaiheet eivät kuitenkaan ole suoria erillisiä viivoja vaan suunnittelu ja kehitys ovat yhdessä. Kummatkin polut tukevat toisiaan aktiivisen kommunikoinnin ja palautteen annon muodossa.

Sekä ketterä kehitys että käyttäjäkeskeinen suunnittelu painottavat iteratiivista ja inkrementaalista lähestymistapaa ohjelmistokehityksessä. Tämä mahdollistaa nopean palautteen keräämisen aikaisemmista iteraatioista ja tätä kautta ohjelmiston parantamisen. Iteratiivisuus on avainasemassa hyvän käytettävyyden takaamiseksi. Iterointia jatketaan niin kauan, kunnes käyttäjän tarpeet ohjelmiston suhteen on täytetty ja tarpeeksi hyvä käytettävyys saavutettu. Ketterän käyttäjäkeskeisen lähestymistavan tulisikin edetä lyhyissä iteraatioissa inkrementaalisesti. [Brhel et al., 2015, s. 171-172] Kuva 8 havainnollistaa tämän.

Integroitu viitekehys lähtee nollaiteraatiosta, joka toimii esivaiheena, jossa suunnitellaan vain tärkeimmät ominaisuudet ja alemman prioriteetin ominaisuudet suunnitellaan vasta tulevissa iteraatioissa yhtä aikaa kehityksen kanssa [Brhel et al., 2015, s. 172]. Tämän nollaiteraation keskiössä on projektin vaatimusten kerääminen [Sy, 2007, s. 118]. Tässä iteraatiossa ei tapahdu vielä kehityksaktiviteetteja, vaan ainoastaan suunnittelutöitä. Iteraatiossa suunnittelijat ja kehittäjät toteuttavat tilannetutkimusta, tehtäväanalyysseja ja haastatteluja, samalla tehdään paperiprototyyppejä, suunnittelukortteja ja käyttäjätarinoita

[Silva da Silva et al., 2011, s. 83]. Foxin et al. [2008, s. 67] mukaan, kun esivaiheen käyttöliittymäsuunnitelma on valmis, se annetaan kehitystiimille. Tämän jälkeen kehitystiimi siirtyy vasta viitekehyksen prosesseihin. Ennen ensimmäisen varsinaisen iteraation alkamista pidetään suunnittelutapaaminen, jossa määritellään, mitä ominaisuuksia kehitetään ensimmäisessä iteraatiossa, ja jäljelle jääneet ominaisuudet siirretään tuotteen kehitysjonoon odottamaan [Fox et al., 2008, s. 67]. Ketteryyden inkrementaalinen luonne mahdollistaa suunnittelun halutuille ohjelmisto-ominaisuuksille niiden prioriteetin mukaan. Näin tärkeimmät ominaisuudet voidaan valita aina suunnittelun kohteeksi ennen niiden kehitysiteeraation alkamista. [Brhel et al., 2015, s. 172] Suunnittelutapaamisiin osallistuvat tahot vaihtelevat tiimikohtaisesti, mutta näihin tulisi osallistui sekä suunnittelijoita että kehittäjiä. Kun on päätetty, mitä implementoidaan, kehitystiimi tekee teknisen suunnitelman ja kehitystyö alkaa. [Fox et al., 2008, s. 67]

Varsinaisessa ensimmäisessä iteraatiossa kehitystiimi alkaa implementoimaan suunniteltuja ominaisuuksia. Samaan aikaan suunnittelijat jatkavat tilannetutkimusta, tehtäväanalyysseja ja haastatteluja seuraavaa iteraatiota varten ja arvioivat nykyistä iteraatiota, antaen palautetta kehitystiimin työstä. Kehitystiimi toimittaa tuotoksin suunnittelijoille myös iteraation aikana. [Silva da Silva et al., 2011, s. 83-84] Uusien suunnitelmien tulee aina pysyä vähintään yhden iteraation edellä kehitystä ja kehitysakiviteettien tulee tapahtua rinnakkain suunnittelun kanssa [Brhel et al., 2015, s. 172; Silva da Silva et al., 2011, s. 83 -84]. Kehitystiimi ja suunnittelijatiimi työskentelevät siis samanaikaisesti. Käytännössä suunnittelijat valmistelevat seuraavan iteraation suunnittelupalaveria. Kun kehitystiimi on implementoinut kaikki ominaisuudet, annetaan uusi käyttöliittymäversio suunnittelijoille tarkistusta ja käytettävyydestä varten. [Fox et al., 2008, s. 67] Jatkuvat suunnittelut ja analysoinnit yhdessä kehityksen kanssa mahdollistavat käyttäjän sekä asiakkaan muuttuviin tarpeisiin vastaamisen, joka puolestaan johtaa ohjelmiston parempaan käytettävyyteen ja hyödyllisyyteen [Brhel et al., 2015, s. 172].

Toisessa iteraatiossa suunnittelijat alkavat tekemään prototyyppkejä sekä suunnittelukortteja ja käyttäjätarinoita. Palaute kulkee yhä tiimien välillä iteraatiossa ja suunnittelijat tarkastavat kehittäjien edellisen iteraation käyttöliittymäversion. [Silva da Silva et al., 2011, s. 83-84] Tarkistuksessa varmistetaan, että kehitystiimi noudatti suunnittelijoiden ohjeita, varsinkin jos kehittäjien joukossa ei ole mukana suunnittelijaa. Jos ominaisuus ei läpäise tarkastusta tai käytettävyydestä suunnittelijat antavat sen takaisin uudelleen implementointiin käynnissä olevassa iteraatiossa. Jos ongelma on liian iso, siirretään sen uudelleen implementointi seuraavaan iteraatioon. [Fox et al., 2008, s. 67-68] Suunnittelijat voivat käyttää ongelmakortteja koodiongelmien raportoimiseksi, jotta kehitystiimi voi näin tehdä tarvittavat korjaukset. Toisessa iteraatiossa suunnittelijat toteuttavat myös tilannetutkimusta, tehtäväanalyysseja ja haastatteluja N iteraatiota varten. Iteraatio N:llä

viitataan viimeiseen iteraatioon. Ennen viimeistä iteraatiota suunnittelijat arvioivat nykyistä iteraatiota ja tekevät käytettävyydestä valmiille suunnitelmille, jotka toteutettiin ensimmäisessä iteraatiossa ja on hyväksytty. Viimeisessä iteraatiossa arvioidaan nykyistä iteraatiota ja tehdään käytettävyydestä. Toiseksi viimeisen iteraation ongelmakorttien ongelmat korjataan ja viimeisessä iteraatiossa havaitut ongelmat korjataan vielä ennen julkaisua. [Silva da Silva et al. 2011, s. 83-84]

Kehityksen iteratiivisten piirteiden takia, kokonaiskuvaa ohjelmistoprojektista on välillä vaikea hahmottaa. Tämä on ongelma jo pelkästään ketterässä kehityksessä ilman käyttäjakeskeisyyden integrointia. Kokonaiskuvan mielessä pitämisen helpottamiseksi on suositeltavaa jakaa dokumentit, välineet ja erityisesti kaikki tieto tiimin jäsenten kesken. Prototyyppien ja suunnittelukorttien käyttö tapaamisissa sekä ongelmakorttien käyttäminen käytettävyysohjelmien raportointiin ovat hyviä käytänteitä integraation kokonaiskuvan ylläpitämiseksi. [Silva da Silva et al., 2011, s. 83]

3.3.2 Etukäteen suunnittelu

Ketterät menetelmät eivät suosi etukäteen suunnittelua, koska ketteryydellä pyritään vastaamaan muuttuviin projektin vaatimuksiin nopeasti. Etukäteen suunnittelemista karteetaan ketteryydessä erityisesti siksi, koska menetelmässä pyritään useisiin julkaisuihin nopeasti uusine ominaisuuksineen. Tämä vaikeuttaa suunnittelijoiden työskentelyä toteuttaa käyttäjätutkimusta ja muita suunnitteluaktiviteetteja ennakkoon. Puutteellisesta suunnittelusta puolestaan voi seurata käyttöliittymä, joka on irtonainen ja epäjohdonmukainen. [Salah et al., 2014, s. 5] Yhtenä merkittävänä tekijänä onnistuneessa yhdistämisessä nähdään kuitenkin etukäteissuunnittelu (eng. design up front), joka tehdään ennen kehitystä. Vaikka laajamittainen etukäteissuunnittelu (eng. big design up front) on ketteriä periaatteita vastaan, niin pientä etukäteissuunnittelua (eng. little design up front) tarvitaan, jotta käyttäjakeskeinen suunnittelu voidaan yhdistää onnistuneesti ketterään ohjelmistokehitykseen. [Brhel et al., 2015, s. 170] Lisäksi Fox et al. [2008] kertoivat laadullisessa tutkimuksessaan, jossa haastateltiin kymmentä ketterässä kehityksessä työskentelevää käyttäjakeskeisyysasiantuntijaa, että suunnittelutoimet aloitettiin ennen itse varsinaisen kehitystyön alkamista. Etukäteissuunnittelu on välttämätöntä, jotta suunnittelija pystyy luomaan yhtenäisen ja johdonmukaisen käyttöliittymän sekä navigointirakenteen. Lisäksi etukäteissuunnittelu auttaa suunnittelijoita löytämään sopivat suunnittelukonseptit alusta alkaen. [Brhel et al., 2015, s. 171]

Etukäteen suunnittelu voidaan aloittaa esivaihesuunnittelulla nolla syklin (eng. cycle zero) tai nolla sprintin (eng. sprint zero) avulla, joissa kummassakin yksi iteraatio käytetään pelkästään suunnitteluun ennen varsinaisen kehityksen alkamista. Tuotteen löytäminen

(eng. product discovery) ja tuotteen luominen (eng. product creation) pitäisi siis erottaa toisistaan alussa. [Brhel et al., 2015, s. 170-171] Tämä esivaihesuunnittelu on yleensä kahden viikon mittainen jakso, jossa on oleellista saada kattava ymmärrys loppukäyttäjistä ja tunnistaa heidän tavoitteet koko projektia ajatellen käyttäjätutkimuksen avulla [Najafi ja Toyoshiba, 2008, s. 533]. Käyttäjän tarpeet ja tavoitteet tulee määrittää tarkasti heti alussa, jotta saavutetaan hyvä käytettävyyden ja hyödyllisyys [Brhel et al., 2015, s. 171]. Etukäteissuunnittelulla estetään myös heikkojen suunnitteluratkaisujen syntymistä, huonoa priorisointia ja epätarkkoja työmääräarvioita [Salah et al., 2014, s. 5]. Chamberlain et al. [2006, s. 151] teettämässä tutkimuksessa selvisi, että suunnittelijoille tuli antaa tarpeeksi aikaa käyttäjätietojen keräämiseen ennen ohjelmointitöitä. Ketteryyden tuomat aikarajoitteet tuo omat haasteensa esivaihesuunnittelulle. Suunnittelijoille jää vähemmän aikaa suorittaa analyysyjä ja testauksia, kuin normaalissa käyttäjäkeskeisessä suunnittelussa. Esivaiheen pituus voi vaihdella paljonkin yrityksestä riippuen aina muutamasta päivästä neljään viikkoon. [Fox et al., 2008, s. 64] Anwar et al. [2014, s. 162] kertoivat tutkimuksessaan, että esivaiheen pituudelle kaksi viikkoa on liian lyhyt ja sen pitäisikin olla jopa neljä viikkoa. Brhel et al. [2015, s. 170] yhdistivät tilannetutkimuksen ja matalan tason prototyypit etukäteissuunnitteluun, koska näiden avulla voidaan kerätä lista vaatimuksia ja tehdä alustavasuunnittelu, jotka annetaan myöhemmin kehitystiimille.

Etukäteissuunnittelua on syytä jatkaa vielä esivaiheen jälkeen niin sanotulla sprintti edellä (eng. one sprint ahead) menetelmällä. Suunnittelijat pitävät suunnitelmat aina yhden iteraation kehitystä edellä. [Silva da Silva et al., 2011, s. 82] Nämä on havainnollistettu aiemman alakohdan kuvassa 8. Etukäteissuunnittelu ei kuitenkaan aina pystytä käyttämään, koska suunnittelijat ovat usein ylityöllistettyjä monien projektien kanssa, jolloin suunnittelu tapahtuu kehityksen kanssa samaan aikaan [Jurca et al., 2014, s. 28].

3.3.3 Käyttäjäkeskeinen suunnittelija osana kehitystiimiä

Käyttäjäkeskeisyyden ja ketterän kehityksen yhdistämisessä kehitystiimin rakenteella on kaksi vaihtoehtoa. Tiimejä voi olla kaksi, joista toinen vastaa suunnittelusta (käyttäjäkeskeinen suunnittelu) ja toinen kehityksestä (ohjelmointi) tai vain yksi tiimi, joka sisältää monitaitoisia suunnittelijoita ja kehittäjiä. Ketterä kehitys suosii yhtä tiimiä, joka koostuu monitaitoisista henkilöistä, joten yhdessä työskentelevän tiimin käyttäminen on sopiva vaihtoehto ketteryyttä ja käyttäjäkeskeisyyttä yhdistäessä. [Brhel et al., 2015, s. 173] Anwar et al. [2014] puolestaan havaitsivat tutkimuksessaan kehitystiimin jakamisen suunnittelijatiimiin ja kehitystiimiin, jotka työskentelivät ristiin toimivana ratkaisuna. Najafin ja Toyoshiban [2008] selvittivät tapaustutkimuksessaan hyvän tiimin sisältävän projektipäällikön, neljä kehittäjää, kolme laadunvarmistajaa ja kolme suunnittelijaa. Yksi suunnittelijoista vastasi käyttäjätutkimuksesta ja testauksesta. Kaksi muuta suunnittelijaa

vastasivat käyttöliittymäsuunnittelusta ja implementoinnista. [Najafin ja Toyoshiban, 2008, s. 532-533]

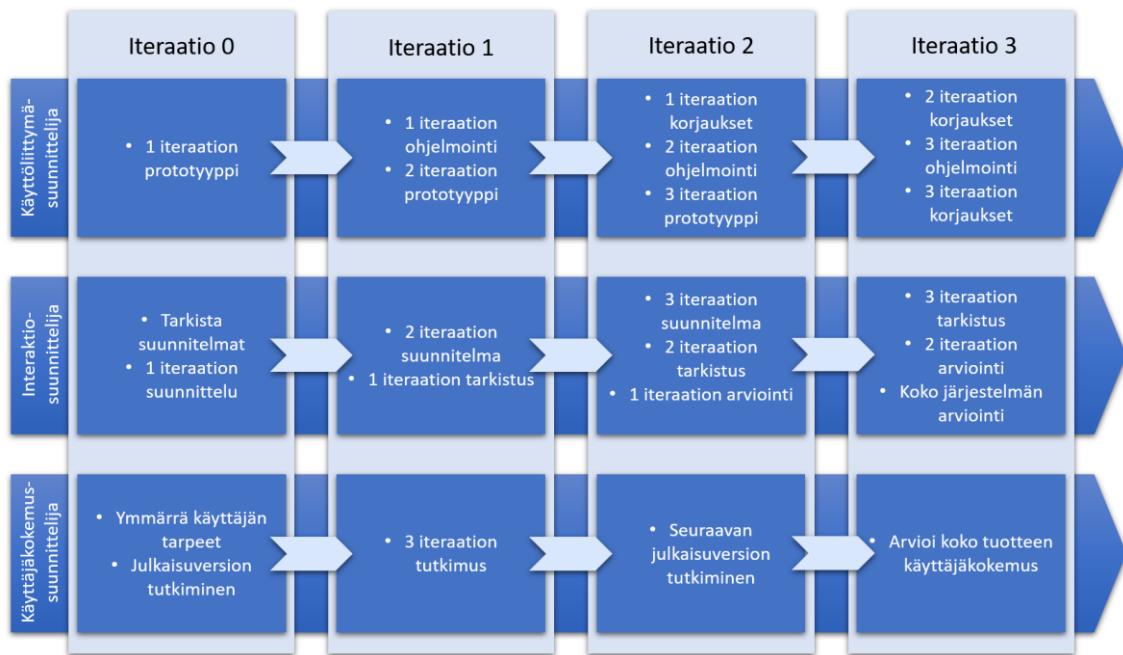
Ketterä kehitys ei keskity käytettävyyteen ja loppukäyttäjiä ei aina pidetä mukana kehitystyössä. Tämä luo usein käytettävyyssongelmia. Kommunikointi suunnittelijoiden ja kehittäjien välillä onkin avainasemassa integroidessa käyttäjäkeskeistä suunnittelua ketterään kehitykseen. [Anwar et al., 2014, s. 161-162] Yhteistyö kehittäjien ja suunnittelijoiden välillä tulisi ylläpitää suunnitteluista kommunikoimalla [Salah et al., 2014, s. 6]. Suunnittelijoiden suunnittelupalaveriisiin ja päiväpalaveriisiin osallistumista pidetään tärkeänä osana ohjelmistokehitysprosessissa, koska tämä edistää kommunikointia ja mahdollistaa käyttäjätutkimustietojen sekä testaustulosten jakamisen. Samalla pystytään korjaamaan väärinymmärryksiä suunnitelmista. [Najafin ja Toyoshiban, 2008, s. 532-533] Chamberlain et al. [2006, s. 151] kertoivat tutkimuksessaan, että kehittäjien sekä suunnittelijoiden tulee kommunikoida ja tehdä töitä yhdessä päivittäin. Kommunikointiongelmia alkaa ilmetä varsinkin, jos kehitystiimin jäsenet eivät osallistu johonkin projektin vaiheeseen tai työtapaan, joten kaikkien tiimin jäsenten tulee olla käytettävissä koko projektin ajan [Chamberlain et al., 2006, s. 152]. Lisäksi jatkuvaa kommunikointia tulee ylläpitää, jotta vältetään myöhästymisiltä ja pullonkauloilta kehitysprosessin aikana [Salah et al., 2014, s. 5].

Yhteistyön ja kommunikoinnin parantamiseksi suunnittelijoilla ja kehittäjillä tulisi olla yhteinen työskentelytila. Tämä yhteistyökulttuuri rohkaisee kehittäjiä näkemään käyttäjäkeskeisen suunnittelun yhtä tärkeänä osana ohjelmistokehitystä, kuin muutkin kehitysvaiheet. Tiivis yhteistyö tiimin kesken on koko kehityksen eduksi, koska se edesauttaa mahdollisten konfliktitilanteiden hoitamista. [Jurca et al., 2014, s. 29] Brhel et al. [2015, s. 177] kertoivat myös tutkimuksessaan, kuinka suunnittelijan tulisi työskennellä samoissa tiloissa kehittäjien kanssa paremman kommunikoinnin takaamiseksi. Lisäksi Williamsin ja Fergusonin [2007, s. 4] tapaustutkimukseen osallistuneet kehitystiimin jäsenet pitivät yhteistä työskentelytilaa tärkeänä, koska tämä helpotti avun pyytämistä tiimin jäseniltä aina tarvittaessa.

Jos suunnittelijat eivät kommunikoi kehittäjien kanssa, on vaarana, että kehittäjät ymmärtävät suunnitteluratkaisut eri tavalla ja näin suunnitelmat implementoidaan väärin. Najafin ja Toyoshiban [2008] toisessa tapaustutkimuksessa kävi juuri näin. Kun projektin kehitys eteni väärin implementoiduilla suunnitteluratkaisuilla, kaikki työ piti käyttää jo luotujen ominaisuuksien korjaamiseen. Aikaa ei annettu enää yhtään käytettävyystestauksille tai suunnitelmien parantamiseen ja lopulta ketterästä prosessista jouduttiin kokonaan luopumaan. Julkaistu lopputuote ei tarjonnut parempaa käyttökokemusta, kuin edellinen versio, vaikka se saavutti alkuperäiset tavoitteet. [Najafi ja Toyshiba, 2008, s. 536]

Käyttäjien tarpeiden ymmärtäminen on tärkeää hyvän suunnittelun kannalta, jonka takia kaikkien kehittäjien tulee ymmärtää ja suostua käyttäjäkeskeisen suunnittelun periaatteisiin, jotta suunnittelijat voivat ylläpitää suunnitteluvision ja tehdä näin päätöksiä ominaisuuksista. Paraskaan suunnitteluvision ei ole hyvä, jos sitä ei jaeta koko tiimin kanssa. Suunnitelmat tulee pitää näkyvillä, koska tämä vähentää uudestaan tehtävää työtä ja helpottaa suunnitteluvision implementointia ohjelmistoon. Jaetun suunnitteluvision ymmärtämiseksi voidaan kehittäjiä ottaa mukaan suunnitteluprosesseihin ja jakaa suunnitteluvälineet, kuten prototyypit kehittäjille. Tietynlainen synkronointi mahdollistaisi paremman rinnakkaisen suunnittelu- ja kehitystyöskentelyn. Kumpienkin työaktiviteetit voidaan synkronoida, jos suunnittelijat ja kehittäjät osallistuvat päivittäisiin tapaamisiin ja kommunikoivat keskenään päivittäin selventääkseen suunnitelmia sekä tiedottaakseen kehittäjiä kaikista muutoksista käyttöliittymään. [Salah et al., 2014, s. 6]

Suunnittelijat vastaavat käyttäjäkeskeisen suunnittelun monista eri suunnitteluaktiviteeteista ja nämä jakautuvat vielä eri rooleihin. Suunnittelijan roolit voidaan jakaa käyttäjäkeskeisessä ketterässä kehityksessä käyttäjäkokemussuunnitteluun, interaktiosuunnitteluun ja käyttöliittymäsuunnitteluun (Kuva 9). Käyttäjäkokemussuunnittelija on vastuussa käyttäjien ymmärtämisestä erilaisien tekniikoiden avulla, kuten käyttäjätutkimuksen, kilpailija-analyysien ja käyttäjien profiloinnin. Interaktiosuunnittelija puolestaan on vastuussa ohjelmiston interaktioiden suunnittelusta ja arvioinneista. Suunnittelua ja arviointia tehdään prototyyppien avulla sekä kehitettävän ohjelmistoversion. Tässä roolissa tulee osata tehdä nopeita prototyyppejä, tuotesuunnittelua, käytettävyystestejä ja myös käyttäjäkokemussuunnittelua. [Silva da Silva et al., 2013, s. 605-607] Anwar et al. [2014, s. 165] painottivat tutkimuksessaan interaktiosuunnittelijan roolia, koska hän pystyy parhaiten selventämään kehittäjille ohjelmiston toiminnallisuuksia. Viimeinen suunnittelijan rooli on käyttöliittymäsuunnittelija, joka suunnittelee graafiset elementit ja kehittää graafisen käyttöliittymän. Käyttöliittymäsuunnittelijan tulee osata tehdä nopeita käyttöliittymäprototyyppejä, yhteissuunnittelua, visuaalista suunnittelua ja muuta käyttöliittymäsuunnittelua. Rooleja suunnittelussa on monia, mutta on tärkeää muistaa, että yleensä ohjelmistoprojekteissa on mukana vain yksi suunnittelija, jolla on useita eri rooleja. [Silva da Silva et al., 2013, s. 605-607]



Kuva 9. Suunnittelijan eri roolit iteraatioissa. Kuva muokattu lähteestä [Silva da Silva et al., 2013, s. 608].

Suunnittelijoilla tulisi olla riittävästi aikaa ohjelmistoprojekteissa, jotta he voivat keskittyä suorittamaan kaikkia heille määrättyjä tehtäviä, joita voi projektikohtaisesti olla useita erilaisia. Suunnittelijoiden laittaminen harvempaan projektiin tai ainoastaan yhteen parantaa tuotteen käyttäjäkokeutusvisiota, koska suunnittelijan ei tarvitsisi venyä useaan eri projektiin samaan aikaan. Tämän lisäksi projekteissa suunnittelijoiden täytyy usein olla muissa rooleissa, kuten interaktiosuunnittelijana ja käyttöliittymäsuunnittelijana. Suunnittelijan työtehtäviin voi lisäksi kuulua käyttäjätutkimusta, markkinointitutkimusta, prototyyppien tekemistä, käytettävyyssarviointeja, käyttäjätestauksia, visuaalista suunnittelua ja ohjelmointia. [Jurca et al., 2014, s. 28-29] Monien roolien takia onkin erityisen tärkeää, että suunnittelija on kehitystiimin täyspäiväisenä jäsenenä [Silva da Silva et al., 2013, s. 607].

Williamsin ja Fergusonin [2007] tekemässä tapaustutkimuksessa selvisi, että edes yksi suunnittelija ei riittänyt yhtä ohjelmistoprojektia kohden suuren työtaakan takia. Suunnittelijan tuli tehdä käyttäjätutkimusta ja samalla suunnittelutuotoksia, kuten prototyyppejä. Toimivana mallina havaittiin suunnittelijoiden jako kahteen rooliin: käyttäjätutkimus-suunnittelija ja prototyyppisuunnittelija. Käyttäjätutkimussuunnittelija vastasi käyttäjien vaatimusten selvittämisestä, toteutti käyttäjätutkimukset ja teki tiivistä yhteistyötä toisen suunnittelijan kanssa. Prototyyppisuunnittelija nimensä mukaisesti toteutti korkean tason prototyyppit, lisäsi uudet suunnitteluideat, vastasi käyttöliittymän dokumentoinnista ja teki

päivittäistä yhteistyötä käyttäjätutkimussuunnittelijan kanssa. [Williams ja Ferguson, 2007, s. 4]

Isot yritykset voivat palkata yhden suunnittelijan aina yhtä ohjelmistoprojektia kohden tai vain muutamaa, mutta pienemmissä yrityksissä suunnittelija osallistuu usein monen eri tiimin projektiin. Ketterän kehityksen takia tämä voi lisätä ylimääräistä työtaakkaa suunnittelijalle. Suunnittelijat osallistuvat useaan päivittäiseen tapaamiseen ja tiukat ketteryyden tuomat aikamääreet eivät tarjoa tarpeeksi aikaa suunnittelutöiden valmiiksi saamiseen. Ylityöllistetyn suunnittelijan taakkaa on mahdollista helpottaa mentoroimalla kehittäjiä suunnittelutöihin, suoraan vähentämällä suunnittelijoiden töitä ja jakamalla työ kahdelle suunnittelijalle omine rooleineen. [Salah et al., 2014, s. 7] Merkittävät työku-
lttuurimuutokset, jatkuva tuki ja holistinen lähestymistapa täytyy ottaa huomioon, jotta voidaan taata integroinnin onnistuminen pitkällä tähtäimellä. Organisaatiokohtaiset käytänteet ja tavat tuleekin olla kunnossa integroidessa malleja. [Jurca et al., 2014, s. 29] Valtakamppailu suunnittelijoiden ja ohjelmistokehittäjien välillä on myös vaarana varsinkin, kun käyttäjäkeskeinen suunnittelu nähdään enemmän valinnaisena tai toisarvoisena varsinaiseen kehitystyöhön verrattuna [Chamberlain et al., 2006, s. 152; Jurca et al., 2014, s. 28; Salah et al., 2014]. Tilannetta ei helpota se, että ketteriin menetelmiin kuuluu toimivan ohjelmiston pitäminen pääasiallisena edistymisen mittarina. Tämä voi aiheuttaa turhaa kilpailua kehittäjien ja suunnittelijoiden välille varsinkin, kun työskentely tapahtuu rinnakkaisesti. [Salah et al., 2014, s. 5] Monet yritykset eivät myöskään pidä suunnittelijaa kokoaikaisena ketterän kehitystiimin jäsenenä [Silva da Silva et al., 2013, s. 608; Jurca et al., 2014, s. 28]. Tutkimuksista käy ilmi, kuinka suunnittelijaa pidetään enemmän toisarvoisena verrattuna muihin ohjelmistokehityksen jäseniin.

3.3.4 Välineet ja toimintatavat suunnittelussa

Välineet ja toimintatavat ovat oleellisessa osassa sekä ketterässä kehityksessä että käyttäjäkeskeisessä suunnittelussa. Ketterä kehitys keskittyy toimiviin prototyyppeihin, kun taas käyttäjäkeskeisyys korostaa välineitä, joilla dokumentoidaan sidosryhmien tarpeita ja kommunikoidaan suunnitteluideoista. Käyttäjätarinat, persoonat ja skenaariot ovat esimerkkejä, joilla kuvataan sidosryhmien tarpeet suunnittelussa (suunnittelukäsitteet selitetty alakohdassa 2.3.3). Käyttäjäkeskeisessä ketterässä kehityksessä tulisi käyttää konkreettisia ja ajan tasalla olevia välineitä, joilla dokumentoidaan ja kommunikoidaan tuotteita ja suunnittelukonsepteja. Lisäksi näiden tulee olla saatavilla kaikille mukana oleville sidosryhmille. [Brhel et al., 2015, s. 176]

Välineiden pääasiallisena tehtävänä nähdään kommunikointi suunnittelijoiden ja kehittäjien välillä [Brhel et al., 2015, s. 176]. Vaikka välineiden pääasiallinen tavoite on toimia kommunikaatio- ja dokumentointivälineinä, ovat ne myös tärkeitä kehittäjien ja

suunnittelijoiden yhteistyön ylläpitämiseksi ja parantamiseksi. Välineiden ja käytäntöjen tulee olla kevyitä käyttäjäkeskeisessä ketterässä mallissa, jotta ne voidaan mahdollistaa kaikkien osallistujien aikatauluihin ja ovat helposti kaikkien saatavilla. [Jurca et al., 2014, s. 29]

Integroidussa ympäristössä toimintatavat ja välineet voidaan jakaa vaiheisiin: tutkimus, suunnittelu ja arviointi [Anwar et al., 2014, s. 162]. Tutkimusvaiheessa persoonat, skenaariot, haastattelut sekä havainnointit ovat tehokkaita tapoja toteuttaa alun tutkimusta. Projektien alussa erityisen tärkeää on tilannetutkimuksen tekeminen eli havainnointia ja haastatteluja loppukäyttäjän käyttökontekstissa. Käyttäjän tekemä työ tulee raportoida ja analysoida huolella ennen suunnitteluvaiheeseen siirtymistä. [Anwar et al., 2014; Brhel et al., 2015] Anwar et al. [2014] ja Brhel et al. [2015] tutkimuksissa tilannetutkimukset selvisivät käytetyimmäksi menetelmäksi tutkimusvaiheessa. Lisäksi Fox et al. [2008, s. 67] ja Silva da Silva et al. [2011, s. 81] tutkimuksissa tutkimusvaiheessa nousi oleelliseksi tilannetutkimus, mutta muista tutkimuksista poiketen myös matalan tason prototyypit. Tilannetutkimus auttoi luomaan ymmärryksen käyttäjästä, jonka jälkeen matalan tason prototyyppi pystytettiin luomaan. Prototyyppi tehtiin iteratiivisesti ja sitä testattiin oikeilla käyttäjillä tai tiimin jäsenillä. Tarkoituksena oli löytää käytettävyysongelmat mahdollisimman varhaisessa vaiheessa ja korjata ne ennen virallisen käyttöliittymäsuunnitelman luovuttamista kehittäjille. [Fox et al. 2008, s. 67] Lisäksi Jurca et al. [2014, s. 29] kertoivat matalan tason prototyyppien kuuluvan käytetyimpiin työkaluihin.

Suunnitteluvaiheessa on erityisen tärkeää käyttää persoonia, piirroksia, rautalankoja ja prototyyppejä [Anwar et al., 2014, s. 162-163]. Näiden käyttäminen tukee suunnitteluideoiden luomista. Prototyyppien käyttö tulisi aloittaa matalan tason prototyypeillä, joita seuraa korkean tason prototyypit. [Brhel et al., 2015, s. 175; Silva da Silva et al., 2011, s. 82] Suunnittelijoiden tulee antaa kehittäjille prototyyppejä ja käyttäjäpalautetta suunnitteluvaiheessa [Chamberlain et al., 2006, s. 151]. Anwar et al. [2014] huomasivat tapaus-tutkimuksessaan, aluksi käytettyjen käyttäjätarinoiden olleen tehottomia suunnitteluvaiheessa. Toimintatavasta luovuttiin, koska prototyypit antoivat tarpeeksi tarkat kuvaukset kehittäjille kehitettävästi toiminnallisuuksista [Anwar et al., 2014, s. 165].

Arviointivaiheessa käytettävyytestaukset prototyypeillä ovat avainasemassa. Nämä voidaan toteuttaa suoraan käyttäjän kanssa tai etätestauksina. Asiantuntija-arviot ovat tärkeitä kunnollisen arvioinnin kannalta. Käytettävyysongelmat tulee tunnistaa, jonka jälkeen korjausehdotukset annetaan kehittäjille. Näidenkin toimintatapojen tulee kuitenkin pysyä kevyinä. [Anwar et al., 2014, s. 162-163]

Brhel et al. [2015] teettämästä kirjallisuuskatsauksesta selvisi käytetyimmät menetelmät ja välineet käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integraatiossa. Tulokset on esitetty taulukossa 8. Tutkimus tukee hyvin muita kirjallisuuskatsauksen tutkimusten tuloksia käytetyistä menetelmistä ja välineistä. Tilannetutkimukset tutkimusvaiheessa, prototyypit suunnitteluvaiheessa ja käytettävyydestä testaukset arviointivaiheessa painottuvat integroidussa ympäristössä.

Taulukko 8. Käytetyimmät välineet ja toimintatavat suunnittelussa. Kaikista käytetyin ensimmäisenä. Taulukko muokattu lähteestä [Brhel et al., 2015, s. 175].

	TUTKIMUSVAIHE	SUUNNITTELUVAIHE	ARVIOINTIVAIHE
VÄLINEET JA MENETELMÄT	<ul style="list-style-type: none">• Tilannetutkimus• Tehtäväanalyysit• Fokusryhmät• Havainnointi• Kyselyt• Haastattelut	<ul style="list-style-type: none">• Prototyypit• Käyttäjätarinat• Piirroksot• Skenaariot• Persoonat• Korttilajittelu• Rautalangat• Mallintaminen• Korttilajittelu	<ul style="list-style-type: none">• Käytettävyydestä testaus• Asiantuntija-arviointi• Valmiskriteeri

Edellä mainittujen käyttäjäkeskeisten välineiden ja toimintatapojen priorisoinnin ja toteuttamisen kanssa on kuitenkin havaittu ongelmia, koska kehittäjät keskittyvät enemmän tekemään toiminnallisia ominaisuuksia kuin käytettävyyttä. Vastuu priorisoinnista tulisi antaa suunnittelijalle ja luoda oma erillinen käytettävyyshälytysjono. Suunnittelu toimintatavoista erityisesti käytettävyydestä järjestelyyn liittyy ongelmia. Ketteryydessä on tarkat aikarajoitteet, jotka vaikeuttavat testien aikataulutusta, joilla arvioidaan ja testataan prototyyppejä loppukäyttäjien avulla. Osa tiimeistä jättääkin testaamisen kokonaan tai testaavat suunnitteluratkaisuja tiimin sisällä. Monien käytettävyydestä testien pitäminen kasvattaa myös budjettia, joka nähdään ongelmallisena. [Salah et al., 2014, s. 5-6]

3.3.5 Sidosryhmien mukana pitäminen

Sidosryhmien jäsenet tulisi pitää aktiivisesti mukana käyttäjäkeskeisessä ketterässä ohjelmistokehityksessä koko kehitysprosessin ajan alusta loppuun asti palautteen keräämiseksi. Ketteryydessä painotetaan ihmiskeskeisyyttä ja sidosryhmien aktiivista mukana pitämistä, mutta tarkkaa jakoa asiakkaan ja loppukäyttäjän välillä ei ole. Ei ole epätavallista, että yksi henkilö täyttäisi kummatkin roolit ketterässä kehityksessä. Tämänkaltaisen henkilö ei voi kuitenkaan edustaa kaikkia sidosryhmiä. Käyttäjäkeskeinen suunnittelu

kuitenkin vaatii suoran yhteyden kehitettävän ohjelmiston loppukäyttäjään. Jotta ohjelmisto saavuttaa sille asetetut vaatimukset on jo aikaisessa vaiheessa pidettävä sidosryhmät mukana kehitysprosessissa. [Brhel et al., 2015, s. 173-174] Brhel et al., [2015, s. 173] tekemän kirjallisuuskatsauksen perusteella sidosryhmien osallistaminen käytettävyyssysteihin on suosituin tapa osallistuttaa käyttäjiä käyttäjäkeskeisessä ketterässä ympäristössä.

Käyttäjä tulisi pitää mukana koko kehitysprosessin ajan, mutta lisäksi kehitystiimin sisällä tulisi olla käyttäjän edustaja, joka voi olla käyttäjän tukena koko kehityksen ajan. Käyttäjän lisäksi asiakkaan tulee olla aktiivinen tiimin jäsen eikä vain passiivinen sivulinen. [Chamberlain et al., 2006, s. 151]

Najafin ja Toyoshiban [2008, s. 534] tapaustutkimuksessa sprinttien ensimmäiset viikot käytettiin käyttäjätestauksien pitämiseen, jossa käyttäjä kokeili aina uutta ominaisuutta muiden jo implementoitujen ominaisuuksien kanssa. Käyttäjät suorittivat kokonaisia prosesseja testeissä, eikä vain ominaisuuskohtaista tehtävää. Tämä mahdollisti käyttäjän tavoitteiden paremman selvittämisen. Kokonaisvaltainen käyttäjätestaaminen edesauttoi kehitystiimiä oppimaan, mitä käyttäjät odottivat ja halusivat ohjelmistolta, jolloin suunnitelmat saatiin vastaamaan käyttäjien tarpeita. Lisäksi tämä helpotti kehitysjonon ominaisuuksien priorisointia. [Najafin ja Toyoshiban, 2008, s. 534] Tämä mahdollistaa tehokkaamman kehityksen ja paremman yhteistyön, joiden avulla tuotetaan parempi ohjelmisto vielä nopeammin [Salah et al., 2014, s. 6].

Fox et al. [2008] tekemän tutkimuksen mukaan kaikki yritykset eivät aina käyttäneet loppukäyttäjiä käytettävyyssysteissä, koska käyttäjät eivät ehtineet testeihin. Jotta välttyään suunnittelun laadun tippumiselta, apuna voidaan käyttää käyttäjätestauksen huolellista valmistelua, heuristista arviointia, matalan tason prototyyppijä ja etänä pidettäviä käytettävyyssystejä. Ketteryyden tiiviiden aikarajojen takia käyttäjiä voi olla vaikea saada juuri oikeaan aikaan käytettävyyssysteihin. Käytettävyyssysteemien järjestämisen varmistaminen vaatiikin etukäteissuunnittelua käyttäjien osallistumisen suhteen, käyttäen hyödyksi tuttuja käyttäjiä, vuokratyöntekijöitä tai etäkäytettävyyssystejä. [Salah et al., 2014, s. 6-7] Ajan puutteen takia myös käytettävyyssysteemien analysointi on hankalaa. Käytettävyyssystejä pidetään silti olennaisena osana integraatiota, koska niiden avulla saavutetaan opittavuutta, tehokkuutta ja aikaista virheiden tunnistamista. [Anwar et al., 2014, s. 165]

3.4 Kirjallisuuskatsauksen yhteenveto

Taulukko 9 esittää yhteenvedon kirjallisuuskatsauksen tuloksista. Taulukko sisältää kirjallisuuskatsauksen tutkimukset sekä tutkimuksissa esiintyneet aiheet käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integraation osalta.

Taulukko 9. Kirjallisuuskatsauksen yhteenveto.

	[Anwar et al., 2014]	[Brhel et al., 2015]	[Chamberlain et al., 2006]	[Fox et al., 2008]	[Jurca et al., 2014]	[Najafi ja Toyoshiba, 2008]	[Salah et al., 2014]	[Silva da Silva et al., 2011]	[Silva da Silva et al., 2013]	[Sy, 2007]	[Williams ja Ferguson, 2007]
Integroitu prosessi											
Scrumin käyttäminen	x	x	x			x	x		x		
Suunnittelu- ja kehityspolut yhdessä		x		x		x	x	x		x	
Nopeat iteraation inkrementaalisesti		x		x			x	x		x	
Suunnitelmat yhden iteraation edellä		x		x				x			
Integraation isokuva								x			
Suunnittelija ja kehitystiimi											
Yhtenäinen tiimi		x			x	x	x		x		x
Kehitystiimi ja suunnittelutiimi erikseen	x			x							
Osallistuminen tapaamisiin						x	x				x
Kaksi tai useampi suunnittelija			x		x	x	x	x	x		x
Suunnitteluroolien jakaminen			x	x	x	x	x	x	x		x
Suunnittelija täyspäiväisenä tiimin jäsenenä		x	x		x	x	x		x		x
Aktiivinen kommunikointi	x	x	x		x	x	x	x			x
Yhteiset työskentelytilat		x			x	x					x
Etukäteen suunnittelemisen											
Esivaihesuunnittelu	x	x	x	x	x	x	x	x	x	x	
Esivaiheen käyttäjätutkimuksen painotus	x	x	x	x	x			x	x	x	
Etukäteissuunnittelu	x	x	x	x	x	x	x	x	x	x	
Kehittäjien tuotosten tarkastus				x				x			
Suunnittelun toimintatavat ja välineet											
Välineet osana kommunikointia	x	x					x	x			
Tilannetutkimus (tutkimusvaihe)	x	x		x				x		x	
Prototyypit (suunnitteluvaihe)	x	x	x	x	x		x	x		x	x
Käyttäjättestaukset (arviointivaihe)	x	x	x		x	x	x	x			x
Sidosryhmien roolit											
Loppukäyttäjän mukana pitämisessä	x	x	x		x	x	x	x			
Asiakkaan mukana pitäminen		x	x		x	x	x	x			

Kirjallisuuskatsauksen tulokset voidaan jakaa aihepiirien mukaan seuraavasti: integroitu prosessi, suunnittelija ja kehitystiimi, etukäteen suunnittelu, toimintatavat ja välineet ja sidosryhmien roolit (Taulukko 9). Käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integroidun prosessin suosituimmaksi ketteräksi menetelmäksi selvisi Scrum-menetelmä, mikä ilmeni kuudessa tutkimuksesta [Anwar et al., 2014; Brhel et al., 2015; Chamberlain et al., 2006; Najafi ja Toyoshiba, 2008; Salah et al., 2014; Silva da Silva et al., 2013]. Toimivana integroituna prosessina nähdään suunnittelu- ja kehityspolkujen kulkeminen

yhdessä [Silva da Silva et al., 2011; Sy, 2007; Najafi ja Toyoshiba, 2008; Fox et al., 2008]. Suunnittelun ja kehityksen tulee edetä yhdessä lyhyissä iteraatioissa inkrementaalisesti, niin kauan kunnes käyttäjän tarpeet on saavutettu [Brhel et al., 2015, s. 171-172]. Suunnitelmien tulisi pysyä aina vähintään yhden iteraation kehitystöitä edellä [Silva da Silva et al., 2011, s. 83 -84; Brhel et al., 2015, s. 172]. Integroitu prosessi voi kasvaa isoksi ja vaikeasti hahmotettavaksi, jolloin apuna tulisi käyttää yhteisiä dokumentteja ja välineitä, kuten prototyyppejä kokonaiskuvan hahmottamisen helpottamiseksi [Silva da Silva et al., 2011, s. 83].

Käyttäjakeskeisen suunnittelun ja ketterän kehityksen yhdistämisessä tiimejä voi olla kahden tyyppisiä. Tiimejä voi olla kaksi, joista toinen vastaa kehityksestä ja toinen suunnittelusta. Ketteryys kuitenkin suosii yhtä tiimiä, joka työskentelee yhdessä. Tätä pidetäänkin luontevana tiimiratkaisuna [Brhel et al., 2015, s. 173]. Suunnittelijoiden osallistamista tiimin suunnittelupalaveriin pidetään tärkeänä, koska tämä edistää kommunikointia ja mahdollistaa suunnittelijoiden tiedon jakamisen kehittäjille [Najafin ja Toyoshiban, 2008, s. 532-533]. Suunnittelijoita tulisi olla tiimissä ainakin kaksi, joista toinen vastaa käyttäjätutkimuksesta ja toinen prototyypeistä [Williams ja Ferguson, 2007, s. 4]. Kaksikin suunnittelijaa voi osoittautua liian vähäiseksi projektin koosta riippuen, jolloin tiimissä tulisi olla kolme suunnittelijaa: yksi käyttäjätutkimussuunnittelija ja kaksi käyttöliittymäsuunnittelijaa [Najafin ja Toyoshiban, 2008, s. 532-533]. Jos suunnittelijoita on vain yksi tiimissä, on erittäin tärkeää, että hän on tiimin täyspäiväinen jäsen [Silva da Silva et al., 2013, s. 607]. Suunnittelijoilla tulisi olla riittävästi aikaa suunnitteluaktiviteettien tekemiseen [Jurca et al., 2014; Williams ja Ferguson, 2007]. Suunnittelijat voivatkin olla ylityöllistettyjä, jolloin töitä tulisi jakaa suunnittelijoiden kesken tai jopa kehittäjille [Williams ja Ferguson, 2007]. Tiimin toimimisen kannalta integroidussa ympäristössä tiivis kommunikointi suunnittelijoiden ja kehittäjien välillä on tärkeässä roolissa [Anwar et al., 2014, s. 161-162; Salah et al., 2014, s. 6]. Suunnitteluideat tulee kommunikoida kehittäjille aktiivisella dialogin vaihdolla ja välineitä, kuten prototyyppejä hyväksikäyttäen [Salah et al., 2014, s. 6]. Tämän kommunikoinnin yhtenä peruspilarina nähdään yhteiset työskentelytilat suunnittelijoiden ja kehittäjien kanssa [Brhel et al., 2015, s. 177; Jurca et al., 2014, s. 29; Williams ja Ferguson, 2007, s. 4].

Etukäteissuunnittelu on välttämätöntä onnistuneen käyttäjakeskeisen suunnittelun ja ketterän kehityksen integroinnin kannalta. Vaikka kattava ennakko suunnittelu on ketteriä periaatteita vastaan, on etukäteissuunnittelua pystyttävä tekemään kohtuullisessa määrin, ennen kehitystyön aloittamista, jotta pystytään luomaan yhtenäinen ja käyttäjän tarpeita vastaava ohjelmisto [Brhel et al., 2015; Fox et al., 2008]. Etukäteissuunnittelun esivaiheessa etenkin käyttäjätutkimuksen tekeminen on tärkeää, jotta käyttäjän tarpeet ja tavoitteet voidaan määritellä tarkasti [Brhel et al., 2015; Najafi ja Toyoshiba, 2008].

Menetelmällä estetään myös heikkojen suunnitteluratkaisujen syntymistä, huonoa priorisointia ja epätarkkoja työmääräarvioita [Salah et al., 2014]. Etukäteissuunnittelua tulee jatkaa vielä esivaiheen jälkeen, eikä kehitys saa mennä suunnitelmien edelle [Silva da Silva et al., 2011, s. 82]. Suunnittelijat tarkastavat kehittäjien edellisen iteraation tuotokset ja antavat mahdolliset korjausehdotukset seuraavaan iteraatioon [Fox et al., 2008, s. 67-68; Silva da Silva et al., 2011, s. 83-84].

Kirjallisuuskatsauksesta selvisi integraatiossa käytettävät toimintatavat ja välineet suunnittelussa. Toimintatapojen ja välineiden esiintyminen voidaan luokitella seuraaviin vaiheisiin: tutkimusvaiheeseen, suunnitteluvaiheeseen ja arviointivaiheeseen [Anwar et al., 2014, s. 162]. Selvästi käytetyin toimintatapa tutkimusvaiheessa on tilannetutkimus, jota toteutetaan suunnittelun alussa loppukäyttäjille tehtävien havainnointien ja haastattelujen avulla [Anwar et al., 2014, s. 162]. Suunnittelussa prototyyppien käyttämistä suositetaan ja niiden tulisi alkaa matalan tason prototyypeistä, joita seuraa korkean tason prototyypit [Brhel et al., 2015, s. 175; Silva da Silva et al., 2011, s. 82]. Arviointivaiheessa käytettävyydestäukset prototyypeillä ovat keskiössä. Nämä voidaan toteuttaa suoraan käyttäjän kanssa tai etätestauksina. [Anwar et al., 2014, s. 162-163] Välineiden pääasiallisena tehtävänä katsotaan olevan kommunikointi suunnittelijoiden ja kehittäjien välillä [Brhel et al., 2015, s. 176]. Välineiden ja toimintatapojen käyttö painottuu eri vaiheissa hieman eri tavalla [Brhel et al., 2015]. Välineiden ja käytäntöjen tulee olla kevyitä integraatiossa, jotta ne voidaan mahdollistaa kaikkien osallistujien aikatauluihin ja ovat helposti kaikkien saatavilla [Jurca et al., 2014].

Kaikki sidosryhmien jäsenet tulisi pitää aktiivisesti mukana koko kehitysprosessin ajan palautteen keräämiseksi [Brhel et al., 2015, s. 173-174; Chamberlain et al., 2006, s. 15]. Jatkuva käytettävyydestäminen käyttäjillä mahdollistaa sen, että kehitettävä ohjelmisto vastaa käyttäjän tarpeita [Najafin ja Toyoshiban, 2008, s. 536]. Loppukäyttäjän lisäksi asiakkaan tulee olla aktiivinen tiimin jäsen eikä vain passiivinen sivullinen [Chamberlain et al., 2006, s. 151]. Käyttäjien osallistuminen käytettävyydesteihin voi kuitenkin olla hankalaa ketterien aikarajoitteiden takia [Fox et al., 2008; Salah et al., 2014]. Käytettävyydestien järjestämisen varmistaminen vaatii suunnittelua, käyttäen hyödyksi tuttuja käyttäjiä tai etäkäytettävyydestejä [Salah et al., 2014].

4. Tutkimusmenetelmät

Tässä luvussa käsitellään tämän tutkimuksen tutkimusmenetelmiä. Tutkimuskysymyseen pyritään hakemaan vastausta laadullisen tutkimuksen avulla, joka suoritetaan teema-haastatteluina puolistrukturoidulla haastattelumenetelmällä, jossa haastatteluteemat ja kysymykset muodostuvat edellä tehdyn kirjallisuuskatsauksen pohjalta. Haastateltaviksi valitaan IT-organisaatioiden suunnittelijoita, jotka tekevät työkseen käyttäjäkeskeistä suunnittelua ketterissä ohjelmistoprojekteissa.

Luvussa selvitetään ensin laadullisen tutkimuksen piirteet, jonka jälkeen siirrytään aineiston keruu menetelmiin. Tämän jälkeen luodaan katsaus haastattelusuunnitelmaan, joka pitää sisällään valitut haastateltavat, haastatteluteemat ja haastatteluprosessin. Näiden jälkeen kerrotaan itse haastatteluiden toteumasta ja analysoinnista.

4.1 Laadullinen tutkimus

Tämän tutkimuksen tutkimusmenetelmäksi valittiin kvalitatiivinen eli laadullinen tutkimus. Hirsjärvi et al. [2007, s. 160] tiivistävät laadullisen tutkimuksen todellisen elämän kuvaamiseen ja pyrkimykseen tutkia tutkittavaa ilmiötä mahdollisimman kokonaisvaltaisesti. Tiedon keruussa suositaan ihmisiä ja tutkija luottaa enemmän keskusteluihin tutkitavien kanssa ja omiin havaintoihinsa, kuin mittausvälineillä hankittavaan tietoon. Lähtökohtana on aineiston monitahoinen ja yksityiskohtainen tarkastelu. Aineiston keruussa suositellaan metodeja, joissa tutkittavien näkökulmat pääsevät esille. Tällaisiin metodeihin kuuluu esimerkiksi teemahaastattelu, ryhmähaastattelu ja osallistuva havainnointi. Kohderyhmä aineiston keruussa valitaan tarkoituksen mukaisesti. [Hirsjärvi et al., 2007, s. 160]

Tässä tutkimuksessa pyritään kuvaamaan mahdollisimman kokonaisvaltaisesti käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integrointia, jonka takia laadullinen tutkimusmenetelmä valittiin. Ilmiöstä pyritään saamaan yksityiskohtaista tietoa, joka vaatii syvällistä paneutumista tutkittavien näkemyksiin. Lisäksi kirjallisuuskatsauksessa mukana olleet tutkimukset olivat kaikki laadullisia tutkimuksia, joka luo varmuutta laadullisen menetelmän valitsemiseen. Tutkimuksien laadullisissa menetelmissä oli kuitenkin eroavaisuuksia.

Laadulliset tutkimukset rakentuvat aiemmista tutkittavasta aiheesta tehdyistä tutkimuksista, empiirisistä aineistoista sekä tutkijan omasta päättelystä ja ajattelusta. Määrälliset tutkimukset puolestaan perustuvat pääasiassa kerätyn aineiston pohjalta tehtyihin mitaustuloksiin ja tutkijan omaan ajatteluun. Raja näiden tutkimusotteiden välillä ei kuitenkaan ole täysin rajoitettu. Laadullisessa tutkimuksessa voidaan käyttää esimerkiksi tilastoja ja määrällisessä tutkimuksessa voidaan hyödyntää aineistona tekstejä. Laadulliset ja

määrälliset tutkimukset eroavat niiden erilaisilla tutkimusasetelmilla eli miten laadittuihin tutkimusongelmiin on tarkoitus kerätä aineistoa tiettyjen menetelmien avulla. [Saaranen-Kauppinen ja Puusniekka, 2006]

Tässä tutkimuksessa käytettävää laadullista tutkimusmenetelmää voidaan luonnehtia eri tavoilla. Tutkittavien näkökulmat tulee huomioida ja kokeellista asetelmaa ei ole. Aineistokoot ovat myös suhteellisen pieniä. Aineistonkeruumenetelminä voi toimia muun muassa haastattelut, elämänkerrat, kirjeet ja havainnointi [Saaranen-Kauppinen ja Puusniekka, 2006]. Seuraavaksi luodaan katsaus valittuun aineistonkeruumenetelmään.

4.2 Aineiston keruumenetelmä

Tutkimusaineiston keräämisen lähtökohtana on tutkimusongelma, jonka perusteella valitaan aineiston keruu menetelmät [Saaranen-Kauppinen ja Puusniekka, 2006]. Tässä tutkimuksessa tutkimusongelmana on kuinka käyttäjäkeskeinen suunnittelu ja ketterä kehitys voidaan integroida suunnittelijan näkökulmasta. Saaranen-Kauppinen ja Puusniekka [2006] luettelevat perinteisiksi laadullisen tutkimuksen aineistonkeruumenetelmiksi haastattelut ja havainnoinnin. Haastattelut valitaan tämän tutkimuksen aineiston keruumenetelmäksi, koska tutkimuksessa halutaan korostaa suunnittelijan näkemyksiä, kysymyksessä on tuntematon alue ja halutaan syventää saatavia tietoja. Hirsjärvi et al. [2007] perustelevatkin haastattelun valintaa silloin, kun halutaan korostaa ihmistä tutkimustilanteessa subjektina, tutkijan on vaikea tietää etukäteen saatavien vastauksien suuntaa ja tarkoitus on saada perusteluja mielipiteille.

Haastattelu kuuluu laajalti käytettyihin aineiston keruu menetelmiin myös tietojenkäsittelytieteissä, kun tutkitaan ihmisiin liittyviä asioita. Tutkimushaastatteluissa voidaan erottaa kaksi perustyyppiä: teemahaastattelu ja strukturoituhaastattelu. Teemahaastattelussa haastattelu etenee vapaamuotoisesti haastattelijan määrittelemien aihepiirien pohjalta ja strukturoidussa haastattelussa haastatteliija esittää haastateltavalle kysymyksiä valmiiksi luodusta kysymysluettelosta. [Tiainen, 2014, s. 2]

Erilaisista haastattelutyypeistä parhaiten tähän tutkimukseen sopii teemahaastattelut. Ennalta määritellyt teemat mahdollistavat haastattelun kulun tutkimuskysymyksen aiheissa, eikä haastattelussa ajauduta epäolennaisiin asioihin. Haastattelumenetelmä antaa enemmän tilaa haastateltavan omille ajatuksille ja perusteluille, joilla saavutetaan kokonaisvaltaista ilmiön ymmärtämistä. Lisäksi teemahaastattelu mahdollistaa pureutumisen integraation onnistumiseen, koska strukturoitua rakennetta ei ole. Teemahaastattelulla saatu tieto on luonteeltaan paljon syvempää kuin esimerkiksi strukturoidulla haastattelulla saatu tieto [Tiainen, 2014, s. 2].

Tutkimukseen valittu aineistonkeruumenetelmä eroaa kirjallisuuskatsauksen tutkimuksien menetelmistä, koska vain yhdessä tutkimuksessa oli käytetty haastatteluita aineiston keruu menetelmänä. Tapaustutkimukset ja kirjallisuuskatsaukset olivat suosituimpia. Tutkimuksessa haastattelut mahdollistavat uusien asioiden paljastumisen ja syvän tiedon saamisen. Tämän on oleellista, koska kirjallisuuskatsauksen tutkimukset käsittelevät käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integraatiota yleisellä tasolla, turvautuen tapaustutkimuksiin sekä kirjallisuuskatsauksiin. Syvän tiedon puute oli havaittavaa. Etenkin suunnittelijoiden kuuleminen oli vähäistä ja tutkimukset perustuivat enemmän havaintoihin. Lisäksi Brhelin et al. [2015] tekemän kirjallisuuskatsauksen mukaan integraatiota käsittelevistä 84 valitusta tutkimuksesta suurin osa oli tapaustutkimuksia ja teoreettisia, ja haastatteluilla toteutettuja tutkimuksia oli vain kolme.

4.3 Haastattelusuunnitelma

Haastattelusuunnitelma kohdassa kuvataan, kuinka haastateltavat saadaan ja miten itse haastattelut toteutetaan. Suunnittelu on tärkeässä roolissa, koska se varmistaa aineiston keruun ja tutkimuksen onnistumisen. Hyvin suunnitelluilla ja toteutetuilla tutkimusasetelmilla voidaan tavoittaa monipuolista tietoa ja lisätä ymmärrystä tutkimuksen ilmiöön [Saaranen-Kauppinen ja Puusniekka, 2006].

4.3.1 Haastateltavat

Haastateltavia valittaessa tavoitteena on saada aineisto, joka edustaa tutkimuskohteen pääpiirteitä. Ilmiöön liittyvä käsitysten kirjo tai koko vaihteluväli ovat tavoitteina laadullisessa tutkimuksessa. Parhaiten nämä ovat usein saavutettavissa heterogeenisen haastateltavajoukon avulla. [Tiainen, 2014, s. 18] Aineiston rajausta ja valintaa tulee kaikissa tapauksissa perustella. Haastatteluja suunniteltaessa tulee vastata kysymykseen, miksi juuri kyseisiä henkilöitä valitaan tutkittavaksi? [Saaranen-Kauppinen ja Puusniekka, 2006]

Tämän tutkimuksen tutkimuskysymys on sidottu suunnittelijoihin, jotka tekevät käyttäjäkeskeistä suunnittelua ketterässä ohjelmistokehitysympäristössä, joten haastateltavien kriteereiksi muodostuu seuraavat:

- Haastateltavan tulee olla suunnittelija, jonka työkuvaan kuuluu käyttäjäkeskeistä suunnittelua.
- Haastateltavan tulee työskennellä ohjelmistoyrityksessä ja ohjelmistoprojektien parissa.
- Haastateltavalla tulisi olla työkokemusta käyttäjäkeskeisessä ketterässä ympäristössä vähintään yhden vuoden.

Kriteerien täytyminen varmistetaan ensin haastattelukutsussa (Liite 3), jossa tiedustellaan haastateltavaksi näiden kriteerien täyttäviä henkilöitä. Vielä itse haastattelussa haastateltavalta varmistetaan kriteerien täytyminen osana taustakysymyksiä. Haastateltavia etsitään verkosta löytyvien ohjelmistoyritysten yhteystietojen avulla. Ohjelmistoyritykset rajataan ensisijaisesti Tampereella toimiviin eri kokoisiin yrityksiin.

Laadullisessa tutkimuksessa aineistona voi olla vaikka yhden henkilön haastattelu tai se voi käsittää joukon yksilö haastatteluja, koska tarkoituksena ei ole etsiä tilastollisia säännönmukaisuuksia tai keskimääräisiä yhteyksiä. Aineistoa voidaan kuvata riittäväksi, kun haastatteluissa alkaa toistua samat asiat. [Hirsjärvi et al., 2007, s. 176-177] Haastattelijointa pyritään saamaan kymmenen. Jos saadussa aineistossa huomataan sen puutteellisuus, tullessaan haastatteluja pitämään vielä lisää.

4.3.2 Haastatteluteemat

Kirjallisuuskatsauksen perusteella luotiin haastatteluteemat ja kysymykset, jotta ne vastaisivat mahdollisimman hyvin tutkimusongelmaan. Kirjallisuuskatsauksessa käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integroinnissa esille selvisi seuraavat teemat ja alateemat:

1. Integroitu prosessi
 - Suunnittelu- ja kehityspolut
 - Kokonaiskuva
2. Suunnittelija ja kehitystiimi
 - Rakenne
 - Kommunikointi suunnittelijan ja kehittäjän välillä
 - Suunnittelijan roolit
3. Etukäteen suunnitteleminen
 - Esivaihe
 - Etukäteissuunnittelu
 - Kehitysversioiden tarkastus
4. Suunnittelun toimintatavat ja välineet
 - Yleiset toimintatavat ja välineet
 - Prototyypit
5. Sidosryhmien roolit
 - Loppukäyttäjä
 - Asiakas

Teemat sekä alateemat on perusteltu kirjallisuuskatsauksen yhteenveto kohdassa (kohta 3.4). Muodostetut teemat vastaavat kirjallisuuskatsauksessa muodostuneita aihepiirejä ja näiden keskeisiä sisältöjä. Kirjallisuuskatsauksesta selvisi viisi pääaihetta: integroitu prosessi, suunnittelija ja kehitystiimi, etukäteen suunnitteleminen, suunnittelun toimintatavat ja välineet ja sidosryhmien roolit. Taulukko 9 havainnollistaa nämä aiheet sekä niissä esiintyvän sisällön. Näistä saatiin luotua myös teemat teemahaastatteluun. Näiden teemojen ja alateemojen pohjalta luotiin haastattelurunko, joka on kokonaisuudessaan liitteenä (Liite 4).

4.3.3 Haastatteluprosessi

Haastateltavia tullaan lähestymään sähköpostiviestillä, joka toimii haastattelupyynnönä. Haastattelut pyritään toteuttamaan 14.1.2019-3.3.2019 välisenä aikana. Ensimmäiset kutsut tullaan lähettämään 9.1.2019 ja näitä lähetetään aluksi kymmenen. Jos haastateltavista ei kuulu mitään ja kymmentä haastateltavaa ei ole saatu, tullaan heitä lähestymään viikon päästä puhelimitse. Mahdollisesti voidaan lähettää vielä lisää haastattelukutsuja. Haastattelukutsussa esitetään aivan aluksi kutsu haastateltavaksi ja luodaan yleiskatsaus tutkimukseen. Kutsussa pyritään motivoimaan tutkimukseen osallistumista painottamalla haastateltavan arvokasta kokemusta ja mahdollisuutta osallistua tieteelliseen tutkimukseen. Haastattelukutsu kokonaisuudessaan on liitteenä 3.

Haastattelut pyritään pitämään haastateltavan yrityksen tiloissa tai Tampereen yliopistolla. Tilaksi pyritään saamaan neuvotteluhuone tai muu vastaavaa rauhallinen tila. Jos sopivaa haastattelu-aikaa tai paikkaa ei löydy tullaan ehdottamaan haastattelun pitämistä puhelimitse.

Jokainen haastateltava haastatellaan erikseen ja yksittäinen haastattelutilanne kestää arviolta 45min – 60min. Aivan haastattelun aluksi toivotetaan haastateltava tervetulleeksi haastateltavaksi ja kerrotaan lyhyesti tutkimuksesta. Tämän jälkeen kysytään nauhoituslupa. Myönnetyn äänitysluvan jälkeen laitetaan äänitys päälle, johon käytetään kännykän, että kannettavan tietokoneen ääninauhureita, jotta äänitys onnistuu varmasti. Samalla haastattelijalla on paperilla teemat kysymyksineen, joiden avulla keskusteltuja aiheita voidaan seurata ja ohjata keskustelua. Haastattelut alkavat helpoilla kysymyksillä, jotka ovat pääasiassa taustatietokysymyksiä. Näin vähennetään haastateltavan jännitystä, kun haastateltava pääsee ääneen helpoilla kysymyksillä. Alkukysymyksien jälkeen siirrytään varsinaisiin kysymyksiin teemojen mukaisesti. Jos kaikkia teemoja ei ehditä käymään lävitse tullaan ehdottamaan vielä uutta haastattelu-aikaa. Haastattelun lopuksi kiitetään vielä haastateltavaa ja kysytään, haluaisiko hän valmiin tutkielman, kun se valmistuu. Haastateltavalta kysytään vielä haluaisiko hän lukea viimeisimmän version tutkielmasta ennen

sen valmistumista. Näin voidaan vielä varmistaa, että haastateltavan kommentit on ymmärretty oikein.

4.4 Haastattelutoteuma

Haastattelut toteutettiin ajalla 22.1.2019 – 15.3.2019. Haastateltavia saatiin yhteensä kahdeksan, joka todettiin sopivaksi määräksi, koska haastatteluissa alkoi toistua samat asiat ja näin saatiin varmuutta aineiston pätevyydelle.

Jokaisen haastateltavan työkuvaan kuului käyttäjäkeskeinen suunnittelu ketterässä ohjelmistokehityksessä. Lisäksi jokaisen haastateltavan yritys oli ohjelmistoalan yritys, jossa suunniteltiin ja kehitettiin ohjelmistosovelluksia. Henkilöstöltään yritysten koko vaihteli aina pienistä suuriin. Yritysten anonymiteetin varmistamiseksi tarkkoja henkilöstölukuja ei julkaista vaan ainoastaan suuntaa antavat luvut ja kokoluokat. Kokoluokitus on taulukossa 10 ja se on tehty hyväksikäyttäen Suomen tilastokeskuksen yrityskokotaulukkoa [Tilastokeskus, 2014]. Taulukossa 10 kerrotaan myös haastateltavatunnisteet. Työkokemusta käyttäjäkeskeisen suunnittelun parissa ketterässä kehityksessä jokaisella haastateltavalla oli ainakin vuoden verran nykyisessä yrityksessä. Lisäksi monet olivat työskennelleet useita vuosia käyttäjäkeskeisessä ketterässä ympäristössä entisissä työpaikoissa, ennen nykyistä työpaikkaansa. Haastateltavien tarkkoja työnkuvia tai työnimikkeitä ei myöskään mainita, jotta haastateltavien anonymiteetti säilyy. Haastateltavien työnkuvat vaihtelivat, mutta kaikilla oli yhteistä käyttäjäkeskeinen suunnittelu ja tähän kuuluva käyttäjäkokemussuunnittelu. Usealla haastateltavalla oli useampi työkuva, kuten käyttöliittymäsuunnittelija ja interaktiosuunnittelija.

Taulukko 10. Haastateltavien yritysten taustatiedot.

Haastateltavan tunniste	Yrityksen henkilöstömäärä	Yrityksen henkilöstö koko
Suunnittelija 1	< 50	pieni
Suunnittelija 2	< 50	pieni
Suunnittelija 3	50 – 249	keskisuuri
Suunnittelija 4	50 – 249	keskisuuri
Suunnittelija 5	250 –	suuri
Suunnittelija 6	250 –	suuri
Suunnittelija 7	250 –	suuri
Suunnittelija 8	250 –	suuri

Suurin osa haastateltavista vastasi haastattelukutsuun jo samana tai seuraavana päivänä, ja haastatteluajankohdat saatiin sovittua hyvissä ajoin. Muutaman haastatteluajankohdan sopiminen kesti, ymmärrettävien työkiireiden takia. Haastattelut toteutettiin Tampereella ja Helsingissä. Haastattelutiloina toimi joko haastateltavan yrityksen neuvotteluhuone tai Tampereen yliopiston neuvotteluhuone haastateltavasta riippuen. Haastateltaville lähetettiin sähköpostitse haastatteluteemat päivää ennen haastattelua, jotta haastattelutilanteista saataisiin sujuvampia. Haastattelut olivat hyvin pitkälti samanlaisia. Teemat elivät hie- man haastateltavasta riippuen, mutta koskaan ei poikettu liian kauas aiheesta. Haastattelut kestivät 35 – 60 minuuttia. Haastattelujen suurimmaksi haasteeksi muodostui vastauksien saaminen kaikkiin kysymyksiin tarpeeksi kattavasti. Osiin kysymyksistä eivät kaikki pys- tyneet vastaamaan yhtä kattavasti kuin toiset, mutta tämä ei haitannut tarpeeksi laajan aineiston saantia.

Kaikki haastateltavat suostuivat haastattelujen tallentamiseen. Haastateltaviin ei tarvinnut olla yhteydessä jälkepäin aineiston osalta, koska nauhoitteissa ei ollut epäselvyyksiä.

4.5 Haastattelutulosten analysointi

Aineiston analysointi aloitettiin keväällä 2019. Aivan aluksi haastattelut litteroitiin, jonka jälkeen jokainen litterointi tarkistettiin vielä kirjoitusvirheiden varalta. Aineistoa kirjoitettiin fonttikoolla 11 ja rivivälillä 1,15 yhteensä 64 sivua. Jokainen litterointi koodattiin haastattelulyhenteillä, kuten: Suunnittelija 1. Näillä koodeilla tullaan yksilöimään sitaatit haastattelutulokset luvussa. Litteroinnin jälkeen luin aineiston useaan kertaan lävitse, jotta se tuli tutuksi ja pystyin löytämään ensimmäisiä yhteisiä aiheita niistä ja vertaamaan poikkesivatko ne ennalta määritellyistä teemoista. Tämän jälkeen siirryin itse analyysivaiheisiin, jotka on kuvattu ja perusteltu alla.

Teemoittelu on luonteva tapa analysoida teemahaastattelun aineistoa. Teemoja eli keskeisiä aiheita muodostetaan yleensä etsimällä aineistosta yhdistäviä tai erottavia seikkoja. Teemat voivat olla samankaltaisia teemahaastattelurungon kanssa, mutta joskus aineis- tosta löytyy myös uusia teemoja. Teemojen muodostamisessa voidaan käyttää apuna koo- dausta. Koodauksessa aineistoon tehdään jäsentäviä merkintöjä, kuten vaikka allevii- vauksia eri väreillä. Näin löydetään keskeiset seikat aineistosta ja näille voidaan miettiä yhdistäviä nimiä eli teemoja. Aineistosta järjestellään kunkin teeman alle ne kohdat, joissa puhutaan kyseisestä teemasta. [Saaranen-Kauppinen ja Puusniekka, 2006] Haastat- telutulosten analysointitavaksi valittiinkin teemoittelu, koska se oli luonteva tapa lähteä analysoimaan teemahaastattelun tuloksia ja aineiston pohjalta ei noussut esille tarvetta käyttää toista menetelmää.

Teemoitteluun voidaan liittää sisältöanalyysi. Sisällönanalyysissä tarkastellaan aineiston yhtäläisyyksiä ja eroja. Analyysin avulla pyritään muodostamaan tutkittavasta asiasta tiivistetty kuvaus, joka linkittää tulokset laajempaan kuvaan ja aihetta koskeviin muihin tutkimuksiin. Sisällönanalyysiä voidaan tehdä teoriaohjaavasti, teorialähtöisesti tai aineistolähtöisesti. Aineiston analyysi ei suoraan perustu teoriaan, mutta linkit siihen ovat havaittavissa. Teoriasta etsitään vahvistuksia ja selityksiä aineistosta tehdyille löydöksille. [Saaranen-Kauppinen ja Puusniekka, 2006] Tässä tutkimuksessa analyysi tehtiin teoriasidonnaisesti, joka on teoria- ja aineistolähtöisen tutkimuksen välimaastossa. Teoriasidonnaisuus muodostui tämän tutkimuksen kirjallisuuskatsauksen tuloksista. Tietty teoriasidonnaisuus valittiin, jotta kerätty aineisto saisi vankemman tuen.

Teemoittelusta saatavien kuvausten sisältöjä voidaan vielä eritellä tarkemmin teemojen kautta tyypittelyssä. Laadullisissa tutkimuksissa ollaan kiinnostuneita, muustakin kuin erinäisistä haastatteluiden kirjoista, jolloin huomio tulee kiinnittää tyypillisyyteen. Tyypittelyllä tarkoitetaan tyypillisten asioiden kokoamista, jolla tarkoitetaan aineiston tiivistämistä havainnollistaviin tyyppeihin. Tuloksena on haastateltavien tarjoamien informaatioaineksien tyypittely. Tyypittelyssä mennään askel teemoittelua vielä pidemmälle, koska siinä kuvataan aineistoa teemojen sijaan vielä laajemmin. Tyyppejä voidaan kuvata koontien koontina eli useat teemat sisältyvät tyyppeihin. [Saaranen-Kauppinen ja Puusniekka, 2006] Tässä tutkimuksessa käytettiin tyypittelyä viimeisenä analysointivaiheena. Tyypittely tuntui sujuvalta tavalta jatkaa analyysiä teemoittelun jälkeen, koska näin saatiin koottua haastattelutulokset tarkemmin ja tulokset saatiin selvemmin näkyviin.

Tässä tutkimuksessa käytettiin analysointiin siis teemoittelua ja tämän tukena teoriasidonnaista sisällönanalyysiä sekä lopuksi tyypittelyä. Tarkoituksena oli löytää tutkimuskysymyksen kannalta oleelliset yhdistävät aiheet ja saada ymmärrystä käyttäjäreunon suunnittelun integroinnista ketterään kehitykseen. Aluksi lähdettiin selvittämään aineistosta löytyviä teemoja ja näiden alateemoja koodausta hyväksikäyttäen. Koodauksessa haastattelutulokset avattiin tekstinkäsittelyohjelmassa ja saman teeman omaavia tuloksia merkattiin samoilla väreillä esiin teksteistä. Aineistosta löytyneet samantyyppiset aiheet linkitettiin toisiinsa ja muodostettiin näkemys teemasta. Tämän jälkeen teemat selitettiin auki ja haastattelulainauksia käytettiin teemojen tukemisessa. Teemojen alle kerättiin kaikki vastaukset, joissa käsiteltiin kyseistä teemaa. Samalla verrattiin kerättyä aineistoa jo tässä tutkimuksessa tehdyn kirjallisuuskatsauksen tuloksiin, näin luoden vankempaa pohjaa tehdyille johtopäätöksille aineistosta. Lopuksi analyysiä seurasi vielä tyypittely, jossa teemoittelun tulokset koottiin tarkemmin tyyppeihin.

5. Haastattelutulokset

Haastatteluaineistossa näkyi monipuolisesti erilaiset tavat, kuinka käyttäjäkeskeinen suunnittelu on implementoitu ketterään kehitykseen. Aineistosta löytyi selviä samankaltaisuuksia, mutta myös mielenkiintoisia eroavaisuuksia ja erilaisia huomioita liittyen menetelmien integrointiin. Aineisto analysoitiin teemoittelun ja tyypittelyn avulla, kuten edellisessä luvussa todettiin. Seuraavaksi esitellään teemoittelun kautta syntyneet teemat ja näiden alateemat. Teemoittelun jälkeen esitetään tyypittelyn tuloksena syntyneet tyypit, jotka jäsentävät teemoittelun tulokset tyyppeihin.

5.1 Teemat

Teemoittelun kautta syntyneet teemat pysyivät lähes samoina verrattuna alkuperäiseen teemahaastattelurunkoon, mutta muutoksia teemoissa oli myös havaittavissa. Haastattelutulosten pohjalta tuli yksi selkeä pääteema lisää: kommunikointi suunnitteluratkaisuista. Tämä oli teemahaastatteluissa vain alateemana, mutta kommunikointi suunnitteluratkaisuista havaittiin aineistosta useaan kertaan eri teemojen alta, joten se oli perusteltua nostaa omaksi pääteemakseen. Muuten teemoissa muutokset olivat pienempiä ja koskivat alateemoja.

Teemoittelua lähdetään esittämään ensin korkealta tasolta käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integraation prosessista. Tämän jälkeen käsitellään suunnittelijaa osana kehitystiimiä. Tätä seuraa etukäteissuunnittelun ja suunnitteluratkaisuista kommunikoinnin tärkeydestä kertominen osana integraatiota. Lopuksi käsitellään käyttäjäkeskeisen suunnittelun toimintatavat ja välineet integroidussa ympäristössä sekä loppukäyttäjän- ja asiakkaan rooli osana ohjelmiston kehittämistä. Teemoittelussa syntyneet 6 pääteemaa ja näiden alateemat ovat seuraavat:

1. Integroitu prosessi
 - Integroidun prosessin piirteet
 - Suunnittelu ja kehityspolkujen eteneminen
 - Integraation kokonaiskuvan mielessä pitäminen
2. Suunnittelija osana kehitystiimiä
 - Ketterä menetelmä
 - Suunnittelijoiden määrä
 - Tiimitapaamiset
 - Suunnittelijan roolit
3. Etukäteen suunnittelu
 - Esivaihesuunnittelu
 - Etukäteissuunnittelu
 - Suunnittelun laadunvarmistus

- Suunnittelutoimet suhteessa aikaan
- Suunnittelutoimet suhteessa rahaan
- 4. Kommunikointi suunnitteluratkaisusta
 - Työskentelytila
 - Dialogin käyminen
 - Menetelmät ja välineet osana kommunikointia
- 5. Suunnittelun toimintatavat ja välineet
 - Tutkimusvaiheen toimintatavat ja välineet
 - Suunnittelu- ja arviointivaiheen toimintatavat ja välineet
- 6. Sidosryhmät
 - Loppukäyttäjä
 - Asiakas

Seuraavaksi käsitellään yllä mainitut teemat. Teemojen esittelyn yhteydessä on nostettu esiin sitaatteja haastatteluista, jotka kuvaavat analyysituloksia. Lisäksi kirjallisuuskatsauksen tuloksia verrataan haastatteluaineistoon, ja näin vahvistetaan tuloksia tai tuodaan esiin eroavaisuuksia. Olennaista onkin tehdä selväksi aineiston ja teorian yhteys.

5.1.1 Integroitu prosessi

Käyttäjäkeskeinen suunnittelu ja ketterä kehittäminen yhdessä ovat loppujen lopuksi yksi integroitu prosessi. Mielipiteet tästä prosessista kuinka se käytännössä toimii, selvisi keskeiseksi pääteemaksi haastatteluaineistoista. Tämän pääteeman alateemoiksi muodostui integroidun prosessin piirteet, suunnittelu- ja kehityspolkujen eteneminen ja integraation kokonaiskuvan mielessä pitäminen.

Integroidun prosessin piirteet ovat pääteeman ensimmäinen alateema. Haastatellut suunnittelijat kertoivat käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integroidun mallin olevan ainoa varteenotettava vaihtoehto sujuvalle ohjelmistokehitykselle. Menetelmien integrointia pidettiin onnistuneena erityisesti ketterän kehityksen tuoman ketteryyden ansiosta, joka mahdollistaa nopean suunnittelun ja takaisin palaamisen suunnitelmissa, jos jokin asia ei toimi haluamalla tavalla. Integroidussa ympäristössä edetään nopeasti iteraatioissa ja tietyissä kehityspisteissä katsotaan, ollaanko tavoitteisiin päästy. Jos tavoitteisiin ei olla päästy, voidaan nopeasti palata taaksepäin ja tehdä suunnitelmat uudestaan. Näin vältetään myös liiallista turhaa suunnittelua, jos suunnitelmissa joudutaan palaamaan paljonkin taaksepäin. Tähän ketterään periaatteeseen käyttäjäkeskeisen suunnittelun nähtiin sopivan hyvin. Alla olevassa kahdessa lainauksessa haastatellut suunnittelijat kertovat, kuinka käyttäjäkeskeisyys sopii hyvin yhteen ketterän kehityksen kanssa, koska se mahdollistaa nopean kehityksen sekä takaisin palaamisen ja uudelleen kokeilemisen:

”Onko toimiva idea ja jos ei oo niin palataan tarpeeks taaksepä ja koitetaan uudestaan ja siitähän ketterässä kehityksessäkin on kyse, eli mennään eteenpäin ja katotaan päästiinkö pisteeseen, johon aiottiin ja katotaan aina seuraava etappi. Siihen käyttäjäkeskeinen suunnittelu toimii hyvin.” Suunnittelija 1

”Nään ilman muuta hyvänä, että se sulautetaan ja ollaan vikkeliä ja ketteriä.” Suunnittelija 3

Ketteryyden tärkeydestä osana käyttäjäkeskeistä suunnittelua kertoivat Jurca et al. [2014, s. 25] ja Brhel et al. [2015, s. 171-172] tutkimuksissaan haastattelutulosten tapaan. Tutkijat kertoivat, että monet nopeat iteraatiot mahdollistavat usean ohjelmistoarvioinnin ja tätä kautta palautteen mukaisten korjauskehdotusten nopean lisäämisen ohjelmistoon. Tämä on samassa linjassa haastattelutulosten kanssa.

Vaikka käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integraatio nähtiin pääasiassa hyvänä asiana, haastatellut suunnittelijat kertoivat myös sen tuomista haasteista. Haastatellut suunnittelijat osoittivat huolen siitä, kuinka käyttäjäkeskeisyys ja loppukäyttäjät voivat unohtua integroidussa mallissa. Ketterä kehitys ei alun perin ota huomioon käyttäjäkeskeisyyttä, minkä vuoksi sen näkyvyys voi olla vähäistä. Alla olevassa lainauksessa suunnittelija kertoo huolen, kuinka käyttäjäkeskeisyys tahtoo usein jäädä ketterässä kehityksessä:

”Mut se mikä mun mielestä on valitettavasti se näkemys, että se käyttäjäkeskeisyys tahtoo usein jäädä.” Suunnittelija 3

Suunnittelijat ovatkin huomanneet, että oma aktiivisuus tuoda suunnitelmia esiin on avainasemassa, jotta käyttäjäkeskeisyys saadaan näkyviin. Seuraavassa lainauksessa suunnittelija kertoo, että hänen tulee olla itse aktiivinen, jotta käyttäjäkeskeisyys pysyy mukana:

”Näkisin, että et se on tosi paljon omasta aktiivisuudesta ja muusta kiinni, että et miten sitä omaa työtä puskee sinne mukaan joka vaiheeseen.” Suunnittelija 8

Haastattelutulokset ovat linjassa Salah et al. [2014, s. 6] tutkimuksen kanssa, jossa kerrottiin suunnitelmien näkyvillä pitämisestä. Kehittäjiä tulee ottaa mukaan suunnitteluprosesseihin ja heille tulee jakaa suunnitteluvälineitä, kuten prototyyppejä. On tärkeää, että kehittäjät ymmärtävät ja suostuvat käyttäjäkeskeisen suunnittelun periaatteisiin, jotta suunnittelijat voivat ylläpitää suunnitteluvisioita, eikä käyttäjäkeskeisyys unohdu. [Salah et al., 2014, s. 6]

Menetelmien yhdistämisessä korostuu suunnittelijoiden ja kehittäjien väliset erot. Molemmilla on omat mielenmallit, jotka luovat omat haasteensa ohjelmiston kehittämiselle, koska kummatkin osapuolet haluavat pitää kiinni omista ajatuksistaan ja työtavoistaan. Erityisesti kysymys siitä kenelle ohjelmistoa tehdään, on integraation tuoma haaste. Suunnittelijat pyrkivät ajattelemaan käyttäjän näkökulmasta, mutta kehittäjät voivat ajatella, että sovellusta tehdään suunnittelijoille, eikä varsinaisille loppukäyttäjille, koska he kommunikoivat lähinnä suunnittelijoiden kanssa, eikä loppukäyttäjien. Alla oleva lainaus kuvaa, kuinka suunnittelijat ja kehittäjät pitävät kiinni omista olettamuksistaan ja ohjelmiston loppukäyttäjä on epäselvä kehittäjille:

”Molemmat haluaa pitää kiinni siitä ns. omasta ja sit semmonen jatkuva kyseenalaistaminen kenelle me tätä softaa tehdään, tehdäänkö teille vai käyttäjille. Sit semmosten tiettyjen kompromissien löytäminen.” Suunnittelija 6

Suunnittelu- ja kehityspolkujen eteneminen on oma alateemansa. Suunnittelu- ja kehityspolkujen katsottiin kulkevan yhdessä suunnittelu hieman kehityksen edellä. Varsinainen suunnitelmien ohjelmointityö alkaa vasta ensimmäisen suunnitteluiteraation jälkeen luotujen suunnitelmien pohjalta. Tätä suunnitteluiteraatio edellä kulkemista jatketaan koko ohjelmistoprojektin ajan kehityksen kanssa rinnan. Haastatellut suunnittelijat kuitenkin kertoivat, että kehittäjät voivat aloittaa back end -ohjelmoinnin eli rajapintaohjelmoinnin jo ohjelmistoprojektin alussa. Seuraavassa lainauksessa suunnittelija kertoo, kuinka suunnittelu ja kehitys kulkevat yhdessä:

”En nää niitä sillain erillisenä polkuna. Kun on joku juttu joka täytyy tehdä niin se täytyy suunnitella ja kehittää eli molemmat tehdään samaan aikaan.” Suunnittelija 7

Fox et al. [2008], Najafi ja Toyoshiba [2008], Silva da Silva et al. [2011] ja Sy [2007] kehittämät integroidun käyttäjäkeskeisen suunnittelun ja ketterän kehityksen mallit nousivat tätä yhdessä kulkemisen kaavaa, lukuun ottamatta back end -ohjelmoinnin aloittamista projektin alussa. Tutkijat eivät suositelleet aloittamaan mitään ohjelmointitöitä, ennen kun ensimmäiset suunnitelmat olivat valmistuneet. Tämä onkin suurin ristiriita integraation etenemisessä kirjallisuuskatsauksen ja haastatteluaineiston välillä.

Suunnittelu pyrkii kulkemaan kehityksen edellä, mutta haastatteluissa kerrottiin, että välillä kehitys voi saavuttaa suunnittelun ja mennä sen edelle. Näissä tapauksissa kehittäjät pyytävät lisäapuja suunnittelijoilta. Seuraavassa lainauksessa haastateltu suunnittelija kertoo suunnittelutoimien kulkevan kehityksen edellä, mutta joskus kehitystoimet voivat mennä suunnittelun edelle:

”Menee toi suunnittelu askeleen edellä yleensä, että tota ainakin alkuvaiheessa se menee, niin et tarvii olla pikkusen jotain valmiina, mitä voi toteuttaa, mutta jossain kohtaa saattaa käydä niinkin, että toteutus menee pikkusen edellä ja pyytää designia.” Suunnittelija 4

Suunnittelijoiden haastatteluissa kerrottiin myös, että suunnittelu- ja kehityspolut voivat kulkea tietynlaisessa putkessa, johon ohjelmistoprojektit voivat helposti päätyä. Putkessa kulkemisessa suunnitteluvaiheen loputtua alkaa kehittäjien ohjelmointivaihe, jolloin suunnittelu siirtyy tietynlaiseen lepotilaan ja jatkuu vasta, kun ohjelmointivaihe loppuu. Tästä putkimallista pyritään kuitenkin pysymään erossa, koska polkujen yhdessä kulkeminen nähdään tehokkaampana tapana tuottaa ohjelmistoja. Putkimalliin ajautumiseen vaikuttaa erityisesti kehitystiimin henkilökemiat. Seuraava lainaus kuvaa, kuinka suunnittelu- ja kehityspolkuja ei tulisi erottaa, mutta ihmiset vaikeuttavat näiden yhdessä kulkemista:

”Niitä ei pitäisi erottaa (suunnittelu- ja kehityspolut), mutta se ei aina välttämättä oo ihmisten päätettävissä tavallaan siihen vaikuttaa ihmiset.” Suunnittelija 6

Putkessa kulkemista pyritään karttamaan, koska tiivis polkujen yhdessä kulkeminen varmistaa ohjelmiston paremman laadun ja oikean suunnan pitämisen. Alla olevassa lainauksessa suunnittelijaa kertoo, kuinka polkujen tulisi kulkea yhdessä, jotta ohjelmistokehitys pysyy oikeassa suunnassa:

”Haluais et se olis enemmän sellasta käsikädessä menoo, koska sillä pystytään just varmistaan se laatu, ja et ollaanko me menossa oikeeseen suuntaan.” Suunnittelija 8

Integraation kokonaiskuvan mielessä pitäminen oli mukana teemahaastatteluissa ja on myös teemoittelun tuloksena omana alateemana. Osa haastatelluista suunnittelijoista kertoi, että heillä ei ollut mitään formaalia tapaa kokonaiskuvan mielessä pitämiseksi ja, että tämä on lähinnä suunnittelijan omista toimista kiinni. Näistä tärkeänä pidettiin aktiivista kommunikointia sidosryhmien kanssa. Seuraavassa lainauksessa suunnittelija kuvaa, kuinka kommunikointi on tärkeä osa kokonaiskuvan mielessä pitämisen kannalta, mutta formaaleja prosesseja ei ole:

”Kommunikoidaan paljon ja tilanteet muuttuu melkein päivittäin, niin tota se melkein riittää, että se niinku kommunikaatio on olemassa. Ei oo mitään formaalia prosessia.” Suunnittelija 4

Lisäksi haastatteluissa kerrottiin kokonaiskuvan esille tuomista omissa suunnitelmissa. Tämä helpottaa projektikokonaisuuksien hahmottamista. Alla oleva lainaus kuvaa, kuinka kokonaiskuvaa pyritään pitämään mielessä suunnittelijan omien suunnitelmien avulla:

”Koitetaan pitää se kokonaiskuva niissä omissa suunnitelmissa, että tavallaan siinä, että sä ymmärrät oikeesti, mikä se on, mitä tehään.” Suunnittelija 8

Silva da Silva et al. [2011, s. 83] kertoivat tutkimuksessaan kokonaiskuvan hahmottamisen tärkeydestä omien suunnitelmien ja etenkin prototyyppien avulla. Haastattelutulokset ovatkin samassa linjassa kirjallisuuskatsauksen kanssa kokonaiskuvan hahmottamisessa tältä osin.

Osa haastatelluista suunnittelijoista kertoi, että heillä oli käytössä Jira-projektinhallintatyökalu, jolla projektien kokonaiskuvaa pidettiin mielessä. Jiraan merkitään projektin osatehtäviä prioriteetteineen ja näille omat vastuuhenkilöt. Seuraavassa lainauksessa suunnittelija kertoo käyttävänsä Jiraa projektien kokonaiskuvan mielessä pitämisen helpottamiseksi:

”Meillä on Jira käytössä ja sinne luotu eritasosta tämmöstä tehtävälistausta ja niin kun prioriteetin kuvausta ikään kuin.” Suunnittelija 5

Suunnittelijat kertoivat vielä isojen projektien kohdalla käyttöön otettavat erilaiset fyysiset taulut toimistojen seinillä, jotka sisälsivät projektin kannalta oleelliset työtehtävät ja aikataulut. Nämä ovat helposti kehitystiimin nähtävissä ja muokattavissa. Alla olevassa lainauksessa suunnittelija kertoo käyttävänsä Jira- projektinhallintatyökalua, mutta myös projektitauluja isoimmissa projekteissa kokonaiskuvan hahmottamiseen:

”Taitaa siellä Jirassa mennä, mutta sit taas jos on iso projekti niin on isot taulut seinillä, et ihan niinkun saattaa olla postit-lappuja seinät täynnä.”
Suunnittelija 3

5.1.2 Suunnittelija osana kehitystiimiä

Toisena pääteemana on suunnittelija osana kehitystiimiä. Haastatellut suunnittelijat kertoivat ketterien menetelmien ohjaavan kehitystiimin toimintaa. Kehitystiimin koko puolestaan vaihteli projektista riippuen, mutta tiettyjä yleistyksiä suunnittelijat pystyivät tekemään. Alateemoiksi pääteemalla muodostui: ketterä menetelmä, suunnittelijoiden määrä, tiimitapaamiset ja suunnittelijan roolit.

Ketteristä menetelmistä suunnittelijat kertoivat kehitystiimien käyttävän Scrum-menetelmää. Puhdasta Scrum-menetelmää ei kuitenkaan kukaan käyttänyt, vaan Scrumista on otettu raamit kehitystiimille, jossa työskennellä. Scrumiin on nimittäin yhdistetty muita ketteriä menetelmiä ja näiden ominaisuuksia, kuten Lean-ajattelua ja Kanbania. Näin käyttäjäkeskeisyys voidaan integroida paremmin ketterään kehitykseen. Yksi suunnittelijoista kertoi kehitystiimin luopuneen kokonaan Scrumista ja siirtyneen pelkän Kanban-menetelmän käyttöön. Alla olevassa kahdessa haastattelulainauksessa kerrotaan, kuinka Scrumiin on sekoitettu muita ketteriä menetelmiä:

”Ketterämenetelmä on sekotus Scrumia ja kaikki agilet-menetelmät lytättynä yhteen. Varsinaisesti ei scrumia eikä leania, mutta se on jotain siltä väliltä. Sprinteissä kuitenkin pyritään tekemään asioita.” Suunnittelija 2

”Kanbanin kautta juokсутaa asioita ja niin edelleen, mutta se tuntuu siltä, että sitä ollaan muokattu tavallaan Scrumin puitteissa.” Suunnittelija 5

Ketteristä menetelmistä kirjallisuuskatsauksessa käytetyimmäksi selvisi myös Scrum-menetelmän käyttäminen [Anwar et al., 2014; Brhel et al., 2015; Chamberlain et al., 2006; Najafi ja Toyoshiba., 2008; Salah et al., 2014; Silva da Silva et al., 2013]. Tutkimukset eivät kuitenkaan maininneet Scrumin hybridiversioita, joka puolestaan ilmeni haastattelulaineistosta.

Kaksi suunnittelijaa mainitsi, että heillä ei ollut vakiintuneita ketteriä menetelmiä. Menetelmän valintaan vaikutti kehitystiimin omat taustat ja osaaminen sekä asiakas. Vaikka tiettyä menetelmää ei oltu valittuna, kuului ketteryys silti jokaiseen ohjelmistokehitysprojektiin. Seuraavassa lainauksessa kerrotaan ketterän menetelmän vaihtuvuus kehitystiimin tarpeiden mukaan:

”Se (ketterä menetelmä) vaihtelee tiimin mukaan ja henkilöitten mukaan ja asiakkaan mukaan, mutta meillä on ketterät menetelmät käytössä aina projekteissa.” Suunnittelija 4

Suunnittelijoiden määrä kehitystiimeissä vaihtelee ohjelmistoprojektin ja yrityksen koon mukaan. Haastatellut suunnittelijat kertoivat vaihtelevasti, että suunnittelijoita voi olla kehitystiimissä yksi tai kaksi. Ideaaliksi suunnittelijoiden määräksi tiimirakenteen kannalta suunnittelijat kertoivat kaksi suunnittelijaa. Suunnittelijat kertoivat, että jos vakiona oli käyttää vain yhtä suunnittelijaa, otetaan toinen suunnittelija mukaan vain, jos työstettävä projekti oli kooltaan iso. Isoissa projekteissa toista suunnittelijaa pidettiin tärkeänä jo pelkästään projektin vakuutuksen kannalta. Projekti tarvitsee kaksi suunnittelijaa, jotka tietävät projektin läpikotaisin, jos vaikka toinen suunnittelijoista estyy kesken

projektin, niin projektin työstämistä voidaan silti jatkaa. Seuraava lainaus kertoo, kuinka toisen suunnittelijan käyttö on tärkeää isoimmissa projekteissa erityisesti projektin vakuutuksen kannalta:

”Jos se on isompi projekti, niin yritetään ottaa siihen toinen suunnittelija, myös vakuutuksenkin kannalta. Sillon siinä on kaksi, jotka tietää sen projektin syvimmän sydämmen.” Suunnittelija 8

Kaksi suunnittelijaa pystyy työskentelemään tehokkaammin, koska he voivat ideoida ja vaihtaa ajatuksia suunnitelmista paremmin yhdessä. Lisäksi kahden suunnittelijan käyttäminen mahdollistaa suunnitteluroolien jakamisen. Alla olevasta kahdesta lainauksesta huomaa, kuinka kahta suunnittelijaa pidetään hyvänä ratkaisuna, koska se mahdollistaa parisuunnittelun ja sen tuomat hyödyt:

”Tehdään aikapaljon parisuunnittelua, että heitellään ideoita toiselle.”
Suunnittelija 2

”Pystyy niitä ideoita pallotteleen ja sparraan toisiaan, niin sillon saatetaan ottaa toinen enemmän kattoon sitä toiminnallisuutta ja toisella voi olla visuaalinen-puoli vahvempaa, niin tukee sillain hyvin toisiaan.” Suunnittelija 7

Kirjallisuuskatsauksen tutkimukset tukivat myös useamman kuin yhden suunnittelijan käyttämistä. Salah et al. [2014, s. 7] sekä Williams ja Ferguson [2007, s. 4] kummatkin tukivat kahden suunnittelijan pitämistä kehitystiimissä, koska suunnittelijan työtaakka voi olla suuri ja tämä mahdollistaa suunnittelijoiden välisen tehokkaan yhteistyön. Tutkimukset kuitenkin painottivat kahden suunnittelijan pitämistä enemmän suuren työtaakan takia, eikä parisuunnittelun tai projektin vakuutuksen kannalta, kuten haastattelutuloksissa kerrottiin.

Tiimitapaamiset selvisivät haastatteluista alateemaksi. Tiimitapaamiset ja niihin osallistuminen oli vaihtelevaa haastattelusta suunnittelijasta riippuen. Osa suunnittelijoista kertoi, että tapaamisia voi olla päivittäin tai viikoittain projektista riippuen. Scrumin mukaiset päivä- ja viikkopalaverit olivat kuitenkin yleisiä. Päiväpalavereissa käydään Scrumin käytäntöjen mukaan lävitse, mitä ollaan saatu aikaan ja ollaan tekemässä. Viikkopalaverit taas käsittelevät projektien tuotoksia laajemmin ja niille on varattu enemmän aikaa. Alla olevat kaksi lainausta kuvaa, kuinka suunnittelijat osallistuvat Scrumin mukaisiin päivä- ja viikkopalavereihin:

”Meillä on Scrumi elikkä meillä saattaa olla aamulla päivittäin (tapaaminen) tai kerran viikkoon (tapaaminen) riippuu projektista.” Suunnittelija 4

"Dailyt on yleensä aamusin tota retroja pidetään viikon tai kahden viikon välein jossa katotaan, miten on mennyt ja mikä on toiminut ja mitä pitäis parantaa toiminnassa. Ne on ne tyypillisimmät, semmoset säännöllisimmät." Suunnittelija 7

Kirjallisuuskatsauksessa Salah et al. [2014, s. 6] tutkimus painotti erikseen, että suunnittelijoiden tulisi osallistua päivittäin tapaamisiin kehittäjien kanssa. Haastateltujen suunnittelijoiden kertomat Scrumin mukaiset päiväpalaverit ovatkin linjassa tutkimuksen kanssa. Päivittäisiin tapaamisiin osallistuminen mahdollistaa suunnittelijoiden ja kehittäjien työaktiviteettien synkronoinnin ja suunnitelmia voidaan selventää [Salah et al., 2014, s. 6].

Toiset suunnittelijat kertoivat, että heistä ei ole suurta hyötyä kaikissa tiimitapaamisissa, koska monet näistä käsittelevät vain teknisiä asioita ja ovat tärkeitä lähinnä vain kehittäjille. Suunnittelijoiden on helppo vaihtaa ajatuksia kehittäjien kanssa koko projektin ajan, joten muodollisia tapaamisia voi olla vähemmän. Alla oleva lainaus kuvaa, kuinka suunnittelijan ei tarvitse osallistua jokaiseen tiimitapaamiseen, koska asioista voidaan keskustella yhteisissä työtiloissa helposti:

"Saattaa osallistua ja voi olla, että ei osallistu. Niin kun sanoin ollaan kaikki tossa pienessä tilassa, niin joku kehittäjä saattaa kysyä että: hei mitä tälle vois tehdä. Tai sitten suunnittelija voi mennä siihen viereen." Suunnittelija 2

Yksi haastateltu suunnittelija kertoi, että hänen työnsä ei edisty niin paljon yhdessä päivässä kaikissa projekteissa, että hänen tulisi osallistua kaikkiin tiimitapaamisiin. Lisäksi moneen tapaamiseen osallistuminen vie aikaa suunnittelutoimien tekemiseltä. Seuraavassa lainauksessa suunnittelija kertookin, kuinka hän ei näe tarpeellisena jokaiseen tapaamiseen osallistumista.:

"Mun työstämäni asiat ei etene niin paljon päivässä, ei ainakaan jokaisessa projektitiimissä (että tarvitsisi osallistua kaikkiin tapaamisiin)." Suunnittelija 5

Toisaalta suunnittelijan pois jättäminen tai jättäytyminen tiimitapaamisista nähtiin myös haasteena, koska näin suunnittelija voi pudota projektista helpommin. Kehittäjät voivat pitää omia tapaamisia ilman suunnittelijoita etenkin, kun he voivat olettaa tietävänsä suunnitelmat tarpeeksi hyvin. Alla olevassa lainauksessa esitetään huoli siitä, kuinka suunnittelija voi pudota projektista, jos hän ei osallistu kaikkiin tapaamisiin:

"Joo eli se on tosiaan haaste, että sais muutkin tiimin jäsenet ymmärtämään, että jos suunnittelija on mukana kaikessa (tiimitapaamisissa), niin se pysyy mukana perillä kaikessa." Suunnittelija 1

Suunnittelijat kertoivat kaikkiin tapaamisiin osallistumisesta olevan hyötyä, mitä ei välttämättä edes huomata. Tapaamiset, jotka eivät linkity suoranaisesti suunnittelijaan nähtiin mahdollisuutena tunnistaa paikkoja, missä suunnittelua tarvitaan. Nämä jäävät helposti huomaamatta, jos suurimpaan osaan tapaamisista ei osallistuta. Seuraavassa lainauksessa suunnittelija kertoo, kuinka suunnittelijan osallistuttaminen kaikkiin tiimitapaamisiin mahdollistaa suunnittelumahdollisuuksien tunnistamisen:

”Aika usein asiat ei mua koske (tiimitapaamisissa), mutta juuri se auttaa tunnistamaan mahdollisuuksia missä mua tarvitaan ja missä ei.” Suunnittelija 5

Suunnittelijan roolit olivat vaihtelevat haastatteluaineiston perusteella, mutta suunnittelija nähtiin usein moniosajana. Suunnittelija, joka tekee käyttäjäkeskeistä suunnittelua, on usein monessa eri työtehtävässä. Näihin työtehtäviin kuului käyttäjäkokemussuunnittelu, käyttöliittymäsuunnittelu, interaktiosuunnittelu ja myynnintukityöt. Lisäksi front end -ohjelmointi kuului muutaman haastatellun suunnittelijan mukaan suunnittelijan työtehtäviin eli käyttöliittymäohjelmointi. Osa suunnittelijoista vastasi näistä kaikista työtehtävistä. Haastatteluaineiston perusteella pystyttiin suunnittelijan työnkuvat rajaamaan viiteen esiintyvään ryhmään:

1. Käyttäjäkokemussuunnittelu, interaktiosuunnittelu ja käyttöliittymäsuunnittelu.
2. Käyttäjäkokemussuunnittelu ja käyttöliittymäsuunnittelu.
3. Käyttäjäkokemussuunnittelu ja liiketoiminnallinentuki.
4. Käyttäjäkokemussuunnittelu ja front end -ohjelmointi.
5. Kaikki edellä mainitut roolit yhdessä.

Seuraavat alla olevat lainaukset kuvaavat, kuinka suunnittelijalla on useampi kuin yksi rooli projekteissa. Suunnittelijalle on yleistä, että yksi suunnittelija vastaa käyttäjäkokemussuunnittelusta, interaktiosuunnittelusta ja käyttöliittymäsuunnittelusta. Suunnittelija tekee käyttäjä tutkimusta, jonka jälkeen seuraa interaktiosuunnittelu ja kerätyn tiedon pohjalta tehdään ensimmäiset käyttöliittymäkuvat. Ensimmäinen lainaus kuvaa, kuinka suunnittelija vastaa käyttäjäkokemussuunnittelusta, interaktioiden suunnittelusta sekä graafisen käyttöliittymän suunnittelusta:

”Suunnittelen miten homman pitää toimia, sitten seuraavaksi suunnittelen interaktiot ja vielä teen graafisen UI-suunnittelen siihen päälle.” Suunnittelija 1

Suunnittelijan monien roolien lisäksi suunnittelijaa voidaan osallistuttaa myös myynnilliseen työhön. Myyntityöhön osallistuminen auttaa suunnittelutoimien myymistä sekä auttaa asiakasta ymmärtämään suunnittelun tärkeyttä. Seuraava lainaus kertoo, kuinka suunnittelija osallistuu myyjän työhön suunnitteluroolien lisäksi:

”Vahvasti meillä on suunnittelijat mukana myyntityön tai siis myyjien jeesinä myymässä ja vähän siinä myyntivaiheessa auttamassa asiakasta, niin kun sparrailee, onko nää oikeesti sitä mitä sä tarvit.” Suunnittelija 6

Suunnittelija voi olla sananmukaisesti moniosaaja. Moniosaajana suunnittelija vastaa suunnittelutöistä, myynnintuesta ja ohjelmoinnista. Alla oleva lainaus kuvaa suunnittelijaa moniosaajana kaikkine rooleineen:

”Kaikki on semmosia moniosaajia, voi puhua palvelumuotoilusta, että sparraa jopa sitä liiketoimintaa ja semmosta palvelu- ja tuotekehitystä, niin siitä sit ihan UX ja UI suunnitteluun ja iso-osa meistä myös tekee fronttikoodia.”
Suunnittelija 4

Nämä tulokset tukevat hyvin kirjallisuuskatsauksen löydöksiä suunnittelijan rooleista ja työtehtävistä. Silva da Silva et al. [2013] tutkimuksessa havaittiin ketterän kehityksen ja käyttäjäkeskeisen suunnittelun integraatiossa muodostuneen kolme eri suunnittelijan roolia, jotka olivat käyttäjäkokemussuunnittelija, interaktiosuunnittelija ja käyttöliittymäsuunnittelija. Vaikka puhtaita yhden roolin omaavia suunnittelijoita ei haastatteluaineistosta löytynyt, samat työtehtävät ovat havaittavissa. Jurca et al. [2014, s. 28-29] kertoivat tutkimuksessaan suunnittelijoiden monista rooleista, joihin voi sisältyä suunnittelun lisäksi liiketoiminnallistatukea sekä ohjelmointia. Suunnittelija nähdään moniosaajana kirjallisuuskatsauksessa ja haastattelutuloksissa.

5.1.3 Etukäteen suunnitteleminen

Etukäteen suunnitteleminen muodostui omaksi pääteemakseen. Ketterässä käyttäjäkeskeisessä suunnittelussa suunnittelija toteuttaa suunnittelutoimia ennen kehitystä, jotta välttään turhilta kehitystoimilta. Pääteeman alateemoiksi muodostui: esivaihesuunnittelu, etukäteissuunnittelu, suunnittelun laadunvarmistus, suunnittelutoimet suhteessa aikaan ja suunnittelutoimet suhteessa rahaan.

Esivaihesuunnittelu haastattelutulosten perusteella pyritään aina aloittamaan tapaamalla ensin asiakas. Ensimmäisissä tapaamisissa asiakkaan kanssa katsotaan projektin vaatimukset ja tavoitteet. Lisäksi asiakkaan kanssa selvitetään loppukäyttäjät ja heidän alustavat tarpeet kehitettävän ohjelmiston suhteen. Alla olevassa lainauksessa kerrotaan, kuinka suunnittelu lähtee heti aloituspalaverista asiakkaan kanssa, jolloin aletaan selvittämään asiakkaan ja käyttäjien tarpeita:

”Aloituspalaveri missä käydään yhdessä se (projekti) läpi, jonka jälkeen ehkä oma rooli alkaa siitä että perehtyy just siihen, mitä tässä tehdään ja mitä asiakas haluaa ja mitä se loppukäyttäjä tarvii.” Suunnittelija 8

Suunnittelijat pyrkivät näkemään loppukäyttäjiä esivaihesuunnittelussa ja kartuttamaan näin ymmärrystä siitä, mitä pitää suunnitella. Oleellista on olla tekemättä mitään oletuksia ja selvittää tarkasti käyttäjien tarpeet haastattelujen ja havainnointien avulla. Kun ymmärrys käyttäjistä sekä tavoitteista on muodostettu, kerätty tieto järjestetään esimerkiksi mallien avulla. Tämän jälkeen luodaan ensimmäiset konseptit ja suunnitelmat, joiden avulla kehittäjät voivat aloittaa varsinaisen ohjelmointityön. Seuraavassa lainauksessa suunnittelija kertoo, kuinka esivaihesuunnittelussa pyritään heti näkemään käyttäjiä, jotta ei tehdä vääriä oletuksia:

”Yritän aina saada projektin lähtemään sillain, että ei tehdä mitään oletuksia vaan lähdetään katsomaan, mistä on kyse. Mennään kentälle selvittämään, mitä sielä oikeesti tapahtuu.” Suunnittelija 1

Kirjallisuuskatsauksen tutkimukset [Brhel et al., 2015; Chamberlain et al., 2006; Najafi ja Toyoshiba, 2008] painottavat myös esivaihesuunnittelun tärkeyttä ja erityisesti ymmärryksen muodostamista käyttäjistä käyttäjätutkimuksen avulla. Tutkimusvaiheessa ymmärryksen luomiseen käytetään tilannetutkimusta (havainnointi ja haastattelut) [Brhel et al., 2015; Silva da Silva et al., 2011]. Haastatellut suunnittelijat kertoivatkin esivaihesuunnittelun sisältävän havainnointia loppukäyttäjän työympäristössä sekä haastatteluja ymmärryksen muodostamiseksi.

Suunnittelu nähdään alkavan ennen kehitystä, mutta todellisuudessa kehitys aloitetaan jo aikaisessa vaiheessa ensimmäisien suunnittelutoimien jälkeen. Osa haastatelluista suunnittelijoista kertoi, että esivaihesuunnittelujen aikana kehittäjät aloittavat jo työstämään back end -ohjelmointia, koska näin välttyään tyhjäkäynniltä kehityksen puolella. Kehittäjät laittavat ohjelmistoympäristöä pystyyn, kunnes ensimmäiset suunnitelmat ovat valmiita. Seuraava lainaus kuvaa, kuinka kehitystyöt aloitetaan samaan aikaan suunnittelun kanssa esivaiheen suunnittelussa:

”Yritetään toimii mahollisimman sillain limittäin, että ei ois tyhjäkäyntiä kehittäjäpuolellakaan, että he pääsee tekeen arkkitehtuuria ja vähän teknistäpohjaa samaan aikaan.” Suunnittelija 4

Yksi haastatelluista suunnittelijoista kertoi ongelmana liiallisen kehitystyön aloittamisen suunnittelun kanssa samaan aikaan jo esivaihesuunnittelussa. Tämä ongelma ilmeni jo pelkästään back end -ohjelmointia aloittaessa, vaikka käyttöliittymää ei oltu vielä aloitettu ohjelmoimaan. Back end -ohjelmoinnilla voidaan liian aikaisessa vaiheessa kehittää ominaisuuksia, mitä ei loppujen lopuksi tarvita ollenkaan. Alla oleva lainaus kuvaa, kuinka voidaan kehittää turhaa ennen valmiita esivaiheen suunnitelmia:

”Ollaan tehty jo jotain, vaikka tutkimus vaiheessa todetaan, että okei tää ei nyt ollutkaan ja mee näin. Ollaan sillon käytetty hirveesti aikaa back end -koodaukseen ja todetaan, että ei tarvita noita ominaisuuksia siellä.” Suunnittelija 1

Kyseinen tapa aloittaa ohjelmointityöt jo esivaihesuunnittelussa menee täysin ristiin Silva da Silva et al. [2011] ja Brhel et al. [2015] tutkimuksien kanssa, jotka painottavat, että esivaiheessa ei pitäisi tapahtua vielä mitään ohjelmointitöitä vaan keskittyä pelkästään suunnitteluun. Kehittäjät voi puolestaan osallistuttaa suunnittelutöihin esivaiheen ajaksi [Brhel et al., 2015, s. 172].

Haastattelujen perusteella esivaihesuunnittelun kesto voi olla yhdestä viikosta aina kolmeen viikkoon projektin koosta riippuen. Esivaiheen pituuden vaihtelun Fox et al. [2008, s. 64] kertoivat myös olevan muutamasta päivästä jopa neljään viikkoon.

Etukäteissuunnittelua tehdään esivaihesuunnittelun jälkeen, jotta suunnitelmat pysyvät kehityksen edellä. Etukäteissuunnittelu mahdollistaa suunnittelu- ja kehitystoimien sujuvan etenemisen, vaikka esivaiheen jälkeen kumpaakin työtetään samanaikaisesti. Etukäteissuunnittelulla suunnitelmat pyritään pitämään kehityksen edellä, joissa tapahtuu ohjelmointi, kuten suunnittelija toteaa alla olevassa lainauksessa:

”Pyritään tekeen niin, että asiat olis suunniteltu hyvissä ajoin ennen, kun toteutus tulee.” Suunnittelija 5

Myös Fox et al. [2008], Najafi ja Toyoshiba [2008], Silva da Silva et al. [2011] ja Sy [2007] kertoivat tutkimuksissaan, kuinka suunnittelun tulisi kulkea yhden iteraation kehitystä edellä. Tutkimukset tukevat haastatteluaineiston tuloksia etukäteissuunnittelun osalta.

Haastatellut suunnittelijat eivät nähneet isoja muutoksia esivaiheessa käytettyihin menetelmiin ja välineisiin jatkettussa etukäteissuunnittelussa, mutta prototyyppien käyttäminen sekä käytettävyydestien pitäminen korostui piirroksien ja konseptien jälkeen. Prototyyppejä iteroidaan suunnitteluprosessin ajan koko ajan pidemmälle ja koitetaan saada selville paremmin käyttäjien todelliset tarpeet. Prototyyppejä käytetään etenkin osana käytettävyydestauksia. Alla oleva lainaus kertoo, kuinka etukäteissuunnittelua tehdään prototyypeillä alun esivaihesuunnittelun jälkeen:

”Kun konsepti on valmiina niin lähetään tekeen jonkin näköstä prototyyppiä ja kun proto on valmiina pystytään palaamaan asiakkaalle tai yleensä ensin asiakkaalle jossain tapauksissa suoraa käyttäjille.” Suunnittelija 1

Kaksi suunnittelijaa mainitsi, että esivaihesuunnittelun jälkeen etukäteissuunnittelussa suunnittelutyömäärä väheni, koska suurin osa tarvittavista suunnittelutoimista oli ehditty tekemään jo esivaihesuunnittelussa. Esivaihesuunnittelussa jälkeen korostuukin suunnittelijan oma aktiivisuus seurata kehityksen tilaa ja auttaa tarpeen mukaan. Alla olevassa kahdessa lainauksessa haastatellut suunnittelijat kertovat, että suunnitelmat tehdään pitkälti valmiiksi jo esivaiheessa ja suunnittelu voi olla tietynlaisessa taukotilassa:

”Suunnitelmat on sillä lailla valmiit, että asioita voidaan hioa ja sitten käytetään samoja menetelmiä kun esivaiheessa.” Suunnittelija 2

”Usein se on vähän holdissa (suunnittelu) ja tota sitten odotellaan jos tehdään jonkinlaista käyttäjätutkimushommaa kentällä jos lähetään iteroimaan uusia versioita tai parempia versioita.” Suunnittelija 4

Suunnittelun laadunvarmistus on osa etukäteen suunnittelemista. Haastatellut suunnittelijat kertoivat, kuinka suunnittelija itse aktiivisesti seuraa kehityksen tilaa ja tarkistaa kehittäjien tuotoksia. Tämä vaatii aktiivista vuoropuhelua suunnittelijoiden ja kehittäjien välillä. Suunnittelija vertaa kehitysversioita suunnitelmiin ja tekee käytettävyydestä uusia. Seuraavassa lainauksessa kerrotaan, kuinka suunnittelija seuraa itse kehitystä ja uusille ominaisuuksille tehdään käytettävyyss- ja bugitestauksia:

”Se on tosi tärkeä seurata kehitystä ja sen valmistumista ja sitten kun uudet ominaisuudet tulee valmiiksi niin on iso osa mun työtä että mä käytettävyydestään ja bugitestaan.” Suunnittelija 5

Suunnittelijat kertoivat, kuinka kehittäjien odotetaan lähestyvän heitä valmiiksi kehitettyjen ominaisuuksien kanssa. Suunnittelijan lisäksi myös kehittäjiltä odotetaan aktiivisuutta. Seuraavassa lainauksessa suunnittelija kertoo, kuinka kehittäjät kysyvät suunnittelijoilta vastaako kehitetty ohjelmisto tai sen ominaisuus suunnitelmia:

”Yleensä se on silleen että hei sain valmiiksi, halukko kattoo tätä, että vastaako tätä suunnitelmia.” Suunnittelija 2

Yksi suunnittelijoista kertoi Kanban-menetelmän käytöstä osana projekteja, jolloin käytössä on erillinen tarkistusvaihe. Projektin työtehtävät juoksutetaan Kanbanin kautta ja jokainen valmistunut työtehtävä käy tarkistusvaiheen lävitse. Alla olevassa lainauksessa haastateltu suunnittelija kertoo, kuinka Kanbanin avulla suoritetaan tarkastusvaihe:

”Jos kanbania tehdään niin ennenkö taski on valmis niin siihen on tarkastusvaihe, jossa voi tulla kattoon hei nyt mä oon koodannu ton voitko kattoo että se toimii niin ku pitää.” Suunnittelija 7

Silva da Silva et al. [2011, s. 83-84] ja Fox et al. [2008, s. 67-68] kertoivat tutkimuksissaan kehittäjien tuotosten tarkistamisen tärkeydestä, kuinka aina edellisen iteraation kehitysversioita tulee verrata suunnitelmiin iteraation lopussa. Tarkistuksella varmistetaan, että kehittäjät ovat noudattaneet suunnitelmia ja tarvittavat korjausehdotukset voidaan esittää [Fox et al., 2008, s. 67-68]. Haastatteluaineiston perusteella laadunvarmistus painottui kuitenkin suunnittelijoiden ja kehittäjien omaan aktiivisuuteen ilman virallisia standardeja, jollei käytössä ollut Kanbanin tyylistä projektinjuoksuttamista.

Suunnittelutoimet suhteessa aikaan ei ollut alateemana alkuperäisessä teemahaastattelurungossa, mutta haastatellut suunnittelijat kertoivat useaan otteeseen ajan suhteen suunnittelutoimiin. Haastatteluista selvisi, että suunnittelijoilla oli tarpeeksi aikaa toteuttaa suunnittelutoimet, mitkä oli sovittu projektien aloituspalavereissa yhdessä suunnittelijan kanssa. Suunnittelijan osallistaminen aloituspalaveriin onkin tärkeää, koska näin suunnittelija voi itse kertoa, mitä pystytään tekemään missäkin ajassa. Näin vältetään liian tiukilta suunnitteluajakatauluilta. Seuraavat kaksi lainausta kuvaavat, kuinka suunnittelija voi vaikuttaa toteutettaviin suunnittelutoimiin ja tätä kautta ajan riittämiseen:

”Oon pystynyt aina kertomaan asiakkaalle paljon pystytään tekeen eli mulla ei tuu kiire siinä hommassa, että pystyn aika hyvin kertomaan, paljon pystyn tekemään missä ajassa ja tehdään sen verran.” Suunnittelija 1

”Usein me voidaan ite vaikuttaa siihen kun tehdään tarjousta, niin suunnittelija pääsee ite siinä kohtaa jo sanelemaan, että minkä verran tähän menis työtä.” Suunnittelija 4

Kaksi suunnittelijaa mainitsi kuitenkin ajan riittämättömyyden ja kiireen suunnittelutoimien toteuttamisessa johtuen monista eri rooleista ja projekteista. Seuraava lainaus kuvaa, kuinka suunnittelija kokee ajan riittämättömyyden suunnittelutoimien toteuttamiseen sekä aikaeron suhteessa ohjelmointiin:

”No pääasiassa sitä aikaa on aika vähän, että se on ehkä itellekki ollu silleen yllätys siinä kohtaa kun on tullu työelämään, suhteessa design ja sitten niinkun devaus niin siinä on aika suuri ero.” Suunnittelija 8

Jurca et al. [2014] kertoivatkin tutkimuksessaan, kuinka suunnittelutoimiin menevä aika tulisi ymmärtää yrityksissä, koska suunnittelijat eivät välttämättä ehdi tekemään kaikkia suunnittelutoimia kunnolla valmiiksi useiden eri roolien ja projektien takia. Lisäksi Salah et al. [2014] toisessa tapaustutkimuksessa havaittiin suunnittelijan ylityöllistyneisyys, jonka takia kaikkia suunnittelutoimia ei keretty tekemään. Kirjallisuuskatsauksentulokset

ovatkin samassa linjassa osan haastatteluaineiston kanssa suunnittelijan ylityöllistämisestä.

Suunnittelutoimet suhteessa rahaan alateema ei myöskään ollut alkuperäisessä teema-haastattelurungossa. Haastatellut suunnittelijat pitivät budjettia liian rajoittavana tekijänä suunnittelutoimien toteuttamiselle. Suunnittelijat kertoivat, kuinka he haluaisivat tehdä enemmän suunnittelutoimia, erityisesti käyttäjätutkimusta projektien alussa, mutta rajallisen budjetin takia tämän toteuttaminen jää liian lyhyeksi. Käyttäjätutkimusvaihetta pidetään myös työläimpänä ja tärkeänä, koska tällöin luodaan pohja käyttäjakeskeiselle suunnittelulle, joten sille tulisi uhrata enemmän aikaa. Alla oleva lainaus kuvaa, kuinka suunnittelija haluaisi tehdä enemmän suunnittelutoimia, mutta ei rajallisen budjetin takia voi:

”Sen mitä oon suunnittelu tekemäni ehdin tekeen, mutta en välttämättä pysty budjetin takia tekemään kaikkea, mitä haluaisin tehdä.” Suunnittelija 1

Syynä lyhyellä alun käyttäjätutkimukselle on tarve aloittaa varsinainen ohjelmointi mahdollisimman nopeasti, mutta tällöin vaarana on turhien ominaisuuksien kehittäminen. Seuraava lainaus kertoo, kuinka suunnittelutoimien budjetin laiminlyönnillä voidaan kehittää ominaisuuksia, joita käyttäjä ei tarvitse:

”Saatetaan polttaa tyhmälläkin tavalla sitä budjettia koska sitten jos se on sitä, mitä käyttäjät ei halua tai ollaan tehty vääränlaisia olettamuksia.”
Suunnittelija 6

Kirjallisuuskatsauksen tutkimukset eivät kuitenkaan kertoneet rahan suhdetta suunnitteluun, vaikka haastateltavat näkivät tämän rajoittavana tekijänä etenkin alun suunnittelussa. Salah et al. [2014, s. 5-6] havaitsivat ainoastaan käytettävyydestien kasvattavan budjettia usein liikaa, jolloin niitä tehdään vähemmän. Tutkimuksista ei kuitenkaan ilmennyt rahankäytön vaikutusta suunnittelutoimiin tai projektin budjettiin.

5.1.4 Kommunikointi suunnitteluratkaisuista

Yhdeksi keskeiseksi pääteemaksi haastattelutuloksista muodostui kommunikointi suunnitteluratkaisuista. Tämä ei kuulunut teemahaastattelujen yhteydessä vielä pääteemoihin, mutta haastattelutulosten perusteella se erottautui selvästi omaksi teemaksi alateemoineen. Haastatellut suunnittelijat kertoivat kommunikoinnin tärkeyden suunnittelijoiden, kehittäjien, loppukäyttäjien ja asiakkaan välillä. Alateemoiksi kommunikoinnille muodostui: työskentelytila, dialogin käyminen sekä menetelmät ja välineet osana kommunikointia

Työskentelytila muodostui ensimmäiseksi alateemaksi kommunikoinnin osalta. Haastatellut suunnittelijat kertoivat, että suunnittelijat ja kehittäjät työskentelevät samassa tilassa, mikä helpottaa ajatustenvaihtoa suunnittelijoiden ja kehittäjien välillä. Yksi yhteinen tila vähentää myös ylimääräisten tapaamisten pitämistä, koska kommunikointia tapahtuu luontevasti valmiiksi jo yhteisissä tiloissa. Lisäksi yhteisissä tiloissa työskentely vähentää suunnittelutyön yksinäisyyttä, koska monesti suunnittelijoita on vain yksi kehitystiimissä. Kolme suunnittelijaa mainitsivat, että tiimit työskentelevät yleensä asiakkaan tiloissa, eikä oman yrityksen, mikä lisää dialogia suunnitteluratkaisuista asiakkaan suuntaan. Tämä nähtiin selvänä etuna projektinkulun kannalta. Alla olevassa kahdessa lainauksessa suunnittelijat kertovat, kuinka yksi yhteinen tila mahdollistaa nopean yhteydenpidon kehittäjiin:

”Meillä on se hyvä puoli, että meillä on yks iso tila eli avoin ympäristö että pääsee tosi nopeesti kävelemään kehittäjäpuolelle. Sitä pyrinkin hyödyntämään silloin kun on projekti, mitä ollaan jo tekemässä.” Suunnittelija 1

”Ollaan kaikki samassa tilassa. Ja kun tulee joku projekti missä tarvitaan näitä kaikkia (suunnittelijat ja kehittäjät), niin se on tosi joustavaa.” Suunnittelija 3

Aiemmatkin tutkimukset kannattivat yhteisiä tiloja, koska näin pystyttiin selvästi parantamaan yhteistyötä ja kommunikointia suunnittelijoiden ja kehittäjien välillä [Brhel et al., 2015; Jurca et al., 2014; Williams ja Ferguson, 2007]. Haastattelutulokset puoltavatkin hyvin kirjallisuuskatsauksen tuloksia.

Kehitystiimin kehittäjien on myös helpompi kasvattaa luottamusta ja ymmärrystä suunnitteluun, kun suunnittelija on samassa tilassa, tämä taas vuorostaan lisää kommunikointia. Alla olevassa lainauksessa suunnittelija kertoo, kuinka suunnittelijan sijoittaminen yhteisiin tiloihin kehittäjien kanssa edesauttaa luottamuksen luomista ja tätä kautta kommunikointia kehittäjien kanssa:

”Oon ollu näkyvästi läsnä, koska se on hidas prosessi saada kehitystiimi luottoaan ja ymmärtämään asioita designer roolista, että voi tulla pienellä kynnyksellä kysymään (samassa tilassa).” Suunnittelija 5

Ymmärryksen ja kommunikoinnin lisäämistä suunnitteluun kannattivat myös Chamberlain et al. [2006] sekä Jurca et al. [2014] tutkimuksissaan, joissa he kertoivat kehittäjien näkevän suunnittelun toisarvoisena verrattuna varsinaiseen kehitystyöhön, jonka takia kehittäjien ja suunnittelijoiden välille tulee kasvattaa luottamusta entisestään. Tämä yhteistyökulttuuri rohkaisee kehittäjiä näkemään käyttäjäkeskeisen suunnittelun yhtä tärkeänä

osana projektia, kuin muutkin kehitysvaiheet. Lisäksi tiivis yhteistyö koko kehitystiimin kesken edesauttaa konfliktitilanteiden hoitamista. [Jurca et al., 2014, s. 29]

Kaksi haastateltua suunnittelijaa mainitsivat maantieteellisesti hajautetut kehitystiimit, joissa tiimit sisältävät jäseniä eri paikkakunnilta, jolloin tiimin yhteinen fyysinen tila puuttuu. Tällöin turvaudutaan usein etäpalavereihin, kun halutaan kommunikoida koko tiimin kesken.

Dialogin käyminen kehitystiimin jäsenten kesken selvisi alateemaksi haastatteluista. Edellä mainittiin yhteisen työtilan tärkeys kommunikoinnin kannalta, mutta yhteinen työtila ei aina takaa aktiivista dialogia suunnittelijoiden ja kehittäjien välillä. Haastatteluissa suunnittelijat kertoivat, että monesti dialogin käyminen suunnittelijoiden ja kehittäjien välillä painottuu eniten projektin alkuun ja loppuun. Tämä tarkoittaa sitä, että projektissa on selvä ajanjakso, jolloin dialogin käyminen on vähäisempää, joka puolestaan on voinut johtaa siihen, että kehitettävä ohjelmisto ei vastaa uusimpia suunnitelmia. Seuraavassa lainauksessa suunnittelija kertoo, että tiivis kommunikointi suunnittelijan ja kehittäjän välillä on tärkeää, jotta suunnitelmat implementoidaan oikein:

”Vaatii tiivistä kommunikaatiota, että suunnittelija osaa kertoa kehittäjälle, että näin sen tulisi toimia ja näin sitä pitäis käyttää. Paljon vaatii iteraatioita, että istutaan vieressä ja tehdään.” Suunnittelija 7

Suunnittelijat kertoivat, että aktiivinen kommunikointi kehittäjien kanssa on tärkeää, koska käyttöliittymän tulee vastata suunnittelijan näkemyksiä eikä kehittäjän. Ilman aktiivista kommunikointia kehitys voi johtaa siihen, että kehittäjät tekevät omia ratkaisujaan suunnitelmien ohella. Alla olevassa lainauksessa suunnittelija kertoo, kuinka kehittäjät tarvitsevat jatkuvaa dialogia suunnittelijoiden kanssa, jotta kehittäjä ymmärtää, miten käyttäjä käyttää ohjelmistoa:

”Tärkeä että jatkuvaa dialogia tapahtuu, koska devaajat tarvii sen, koska sen käyttöliittymän, miten ne käyttäjät käyttää sitä, niin sen ei tarvi olla pellinalla tapahtuva ilmentymä vaan nimenomaan suunnittelijaa tarvitaan siihen, joka tekee siitä käytettävän.” Suunnittelija 6

Suunnittelijat kertoivat, että dialogin syntyminen on ennen kaikkea kiinni tiimin jäsenistä sekä heidän välisistä henkilökemioista. Tiimin jäsenten tuleekin olla oma-aloitteisia tiedon välittämisessä koko kehityksen ajan. Seuraava lainaus kuvaa, kuinka aktiivista dialogia tulee ylläpitää tiimin jäsenten kanssa ja henkilökemiat vaikuttavat dialogin määrään:

”Sellanen koko aikanen, jatkuva kommunikointi ja vuorovaikutus niiden kaikkien projektijäsenten kanssa on tosi tärkeä, mutta myös se, että se on projektiriippuvaista ja semmonen henkilökemia on tosi vahvassa osassa siinä.” Suunnittelija 8

Aktiivisen dialogin käymisen ongelmana nähtiin tiimin jäsenten uppoutuminen omiin työtehtäviinsä, jolloin kommunikaatio myös katkeaa. Lisäksi haastatellut suunnittelijat kertoivat, kuinka suunnittelija voi unohtua projektin edetessä, jolloin kommunikaatio katoaa. Suunnittelijan taitoja tulisi muistaa hyödyntää koko projektin ajan, eikä olettaa, että suunnitelmat tiedetään tarpeeksi hyvin. Tästä seuraa ongelmia kehityksen jatkuessa, kun suunnitelmat eivät kohtaa enää toteutusta. Alla oleva lainaus kuvaa, kuinka suunnittelija voi unohtua projektin edetessä ja tätä kautta kommunikointi katkeaa:

”Pystyn itse suunnittelemaan sen sillain, että voin viedä viestii itse eteenpäin, mutta jos se jossain vaiheessa katkee siinä matkalla. Tavallaan oletetaan tai tiedetään tarpeeksi, niin sitten se kommunikointi katoaa myös samalla.”

Suunnittelija 1

Vajaan dialogin käymisen varjopuolen kokivat Najafin ja Toyoshiban [2008] tutkimuksessaan, kun kommunikointi suunnittelijoiden ja kehittäjien välillä oli vähäistä, implementoitiin suunnitteluratkaisut väärin ja kaikki työ jouduttiin käyttämään jo luotujen ominaisuuksien korjaamiseen. Jatkuvaa kommunikointia tulee ylläpitää, jotta vältytään pulonkauloilta ja myöhästymisiltä koko kehitysprosessin aikana [Salah et al., 2014, s. 5]. Kirjallisuuskatsauksen tulokset ovatkin linjassa haastattelutulosten kanssa, koska haastatellut suunnittelijat kertoivat huolesta kommunikoinnin katkeamisesta ja näin suunnitelmien väärin implementoinnista.

Menetelmät ja välineet osana kommunikointia kuuluu kommunikoinnin alateemoihin. Haastatellut suunnittelijat kertoivat kehitystiimin pääasiassa itse päättävän käytettävät menetelmät ja välineet kommunikointiin, mutta tiettyjä linjauksia yrityksen sisällä on tehty. Ensimmäisistä suunnitelmista kommunikoidaan kehittäjille yleensä paperihahmotelmien, piirroksien tai muiden kevyiden käyttöliittymäkuvien avulla. Näiden avulla voidaan hyvin havainnollistaa suunnittelijoiden ideat. Hahmotelmat eivät pelkästään kuitenkaan riitä, vaan suunnittelijan itse tulee selittää kehittäjille hahmotelmien suunnitteluratkaisut, jotta vältytään väärinymmärryksiltä. Alla olevassa lainauksessa kerrotaan, kuinka kehittäjille voidaan kommunikoida suunnitelmista paperihahmotelmien avulla:

”Sitten kun päästään vähän eteenpäin niin sit täytyy sinne kehittäjille kertoa, että tällänen pitäis tehdä. Yleensä näyttää niitä paperihahmotelmia, että tämmöstä.”

Suunnittelija 7

Suunnittelijat kertoivat prototyyppien käyttämisen olevan tärkeässä roolissa kommunikoitaessa suunnitteluratkaisuja kehittäjille. Näiden avulla suunnitteluratkaisu voitiin havainnollistaa kehittäjille vielä tarkemmin. Erityisesti interaktiiviset prototyypit helpottivat ohjelmiston toimintojen ja ulkoasun selventämistä kehittäjille. Näiden välineiden hyväksikäyttö vaatii paperihahmotelmien tapaan myös suunnittelijoiden selitykset, jotta kehittäjät eivät tee vääriä johtopäätöksiä suunnitelmista. Seuraavassa lainauksessa suunnittelija kertoo, kuinka prototyypin avulla kommunikoidaan suunnitteluratkaisuista:

”Tehdään jonkin näköinen käyttöliittymä prototyyppi, minkä voi antaa kehittäjälle. Tällänen sen pitäisi olla. Tän pitäisi toimia näin ja tän pitäisi toimii näin.”

Suunnittelija 1

Kirjallisuuskatsauksen tutkimukset olivat samassa linjassa haastatteluaineiston kanssa. Salah et al. [2014] totesivat tutkimuksessaan, kuinka suunnitteluratkaisu sekä suunnitteluvision voi jakaa kehittäjille prototyypin muodossa. Kirjallisuuskatsauksen mukaan välineiden pääasiallinen tehtävä onkin toimia kommunikointivälineinä suunnittelijoiden ja kehittäjien välillä [Brhel et al., 2015, s. 176].

Kommunikointi välineistä toisena merkittävänä pidettiin viestittelysovelluksien käyttämisestä. Näiden ahkera käyttö mahdollistaa nopean viestittelyn tiimin kesken suunnitteluratkaisuista. Suunnittelijat kertoivat käyttävänsä Slack-viestittelysovellusta, jolla voidaan nopeasti kohdistaa viestejä kehitystiimin jäsenille. Pikaviestittelyn avulla voidaan varmistaa suunnitteluratkaisuja ja ilmoittaa pienistä suunnitelmamuutoksista. Alla oleva lainaus kuvaa, kuinka Slack-viestittelysovellusta käytetään osana kommunikointia:

”Slacki on hyvä kun pystyy kirjoitteleen ja pinnaan kaverii (merkitsee osaksi viestiä), kun tarttee.” Suunnittelija 4

Kaksi suunnittelijaa piti hyvänä menetelmänä tapaamisia kehittäjien kanssa ensimmäisistä suunnitelmista, joissa keskustellaan kehittäjien kanssa suunnitelmien toteuttamiskelpoisuudesta. Näin saadaan varmistus siitä, voidaanko suunnitelmia jatkaa vai pitääkö niihin tehdä muutoksia. Seuraavat kaksi lainausta kertoo, kuinka ensimmäisten suunnitteluratkaisujen toteuttamiskelpoisuudesta tulisi puhua kehittäjien kanssa ennen niiden jatko-suunnittelua:

”Tavotteena on, että osallistetaan kehittäjät myös siihen eli katselmois myös suunnitelmia siinä vaiheessa kun tehdään suunnitelmia kommentoidaan onko ne toteuttamiskelpoisa.” Suunnittelija 2

”Suunnittelen jotain, niin sen täytyy olla myös toteutettavaa ja sitten tavallaan sellanen vuoropuhelu siinä vaiheessa.” Suunnittelija 8

5.1.5 Suunnittelun toimintatavat ja välineet

Toimintatapoja ja välineitä käyttäjäkeskeisessä suunnittelussa havaittiin haastatteluista useita, joiden käyttö korostuu ketterässä ympäristössä. Näissä oli paljon hajontaa haastattelusta suunnittelijasta riippuen, mutta prototyyppien käyttäminen sekä näillä tehtävät käyttäjätestaukset painottuivat jokaisessa haastattelussa (suunnittelukäsitteet selitetty alakohtassa 2.3.3). Alateemoiksi pääteemalle muodostui: tutkimusvaiheen toimintatavat ja välineet sekä suunnittelu- ja arviointivaiheen toimintatavat ja välineet. Suunnittelu- ja arviointivaiheen toimintatavat ja välineet muodostettiin yhdeksi omaksi alateemaksi, koska näissä esiintyvien toimintatapojen ja välineiden välillä on päällekkäisyyttä.

Tutkimusvaiheen toimintatavat ja välineet muodostui ensimmäiseksi alateemaksi. Tutkimusvaiheessa korostuu haastattelut, havainnoinnit ja työpajat. Suunnittelun alkuvaiheessa tehdään loppukäyttäjien havainnointia ja haastatteluja, joiden avulla luodaan ymmärrys käyttäjistä. Lisäksi työpajat voivat kuulua alun suunnittelutoimiin haastateltujen suunnittelijoiden mukaan. Työpajoissa käydään keskusteluja suunnitteluratkaisuista ja täytetään suunnittelukanvainta käyttäjien sekä asiakkaiden kanssa. Työpajojen jälkeen suositetaan vielä erikseen haastattelujen pitämistä, jotta saadaan vielä syvempi tieto käyttäjistä ja heidän tarpeistaan. Tarkoituksena on saada tarpeet näkyviksi ja tätä kautta kerätä ymmärrys siitä, mitä tulee suunnitella ja lopulta kehittää. Alla olevassa kahdessa lainauksessa suunnittelijat kertovat, kuinka työpajoja, haastatteluja ja havainnointia käytetään ymmärryksen muodostamiseen loppukäyttäjistä:

”Meillä on ns. työpajoja, jotka ei aina oo lappuleikkejä vaan tavallaan palaverimuotonen, mutta kutsutaan työpajaks, koska keskustellaan eteenpäin ratkasuja. Sen lisäksi käydään haastattelemassa sekä asiakkaan edustajia että asiakkaan alaisia ja muita käyttäjiä. Jonkin verran on observointia.” Suunnittelija 5

”Sit jos työpajan lisäksi syvällisempää, niin haastattelun keinoin siinä pääsee vähän intiimimmin jutteleen pääsee vielä syvemmin, sitten tota sieltä käyttäjästä sitä ymmärrystä.” Suunnittelija 6

Ymmärryksen muodostamiseen Brhel et al. [2015] sekä Jurca et al. [2014] esittivät samoja toimintatapoja ja välineitä. Anwar et al. [2014] ja Brhel et al. [2015] kertoivat tutkimuksissa erityisesti tilannetutkimuksien (havainnointi ja haastattelut) olevan käytetyimpiä menetelmiä integroidussa ympäristössä, aivan kuten haastateltavat suunnittelijat. Tämä on linjassa haastattelutulosten kanssa.

Suunnittelu- ja arviointivaiheen toimintatavat ja välineet on yksi alateemoista. Tässä vaiheessa korostuu persoonat, konseptit, rautalankamallit, prototyypit ja käytettävyydestaukset. Haastatellut suunnittelijat kertoivat suunnitteluvaiheessa käyttävänsä persoonia, konsepteja ja rautalankamalleja, joiden avulla luodaan ensimmäiset versiot ohjelmiston rakenteesta ja toiminnallisuuksista. Seuraavassa lainauksessa suunnittelija kertoo rautalankamallin käyttämisestä toiminnallisuuksien hahmottamiseksi:

”Ekaks rautalankamallit, joilla saadaan enemmän sitä toiminnallisuutta, että voisko se näin, miten se menis ja sit sitä taas testausta.” Suunnittelija 6

Sekä suunnittelu- että arviointivaiheessa prototyyppien käyttäminen korostuu. Jokainen suunnittelija kertoi prototyyppien olevan jossain määrin mukana suunnittelussa. Prototyyppien käyttämistä perusteltiin sillä, että niiden avulla kehittäjät, asiakas ja loppukäyttäjät pääsevät konkreettisesti käsiksi suunnittelijoiden suunnitteluideoihin. Prototyyppijä iteroidaan koko ajan pidemmälle käytettävyydestauksien sekä asiakkaalta saadun palautteen avulla. Prototyyppien käytön hyötyihin suunnittelijat kertoivat kuuluvan ajan ja rahan säästäminen, koska prototyyppijä voidaan tehdä nopeasti ja niiden avulla saadaan hahmoteltua koko ohjelmisto toimintoineen ilman aikaa vievää ohjelmointityötä. Prototyyppien avulla varmistetaan ensin, mitkä ominaisuudet toimivat ja mitkä eivät. Haastatellut suunnittelijat kertoivat prototyyppien kulkevan mukana koko projektin ajan. Lisäksi prototyyppien hyötyihin lueteltiin jo edellisissä alateemoissa mainitut kommunikoinnin helpottaminen ja projektin kokonaiskuvan hahmottaminen. Seuraavissa kahdessa lainauksessa kerrotaan, kuinka prototyypit kulkevat koko projektin ajan mukana ja tarjoaa kehittäjille tavan päästä käsiksi suunnitteluratkaisuihin:

”Ne (prototyypit) kulkee koko ajan mukana. On toivottavaa, että kehittäjä katsois mallia sieltä kun kehitys alkaa.” Suunnittelija 2

”Usein käytössä (prototyypit). Niillä on helppo havainnollistaa sitä työkalun flowta, että miten se etenee, ilman että sitä ois tarvinnu koodata sitä työkalua toimivaksi.” Suunnittelija 7

Myös Anwar et al. [2014], Brhel et al. [2015] ja Jurca et al. [2014] kertoivat tutkimuksissaan prototyyppien kuuluvan käytetyimpiin välineisiin osana suunnittelua, etenkin kun ne tarjoavat tarpeeksi tarkat kuvaukset kehittäjille kehitettävistä toiminnallisuuksista. Haastatteluaineistossakin mainittu prototyyppien kehittäminen läpi projektin on myös linjassa Fox et al. [2008, s. 67] tutkimuksen kanssa, jossa kerrotaan nopean iteratiivisen prototyyppien kehittämisen olevan tärkeä osa integraatiota, koska näin pystytään korjaamaan käytettävyysongelmat nopeasti.

Suunnittelijat kertoivat prototyypeistä suosivansa interaktiivisia prototyyppejä. Tämä interaktiivinen prototyyppi vastaa kehitettävää ohjelmistoa sekä ulkoasun että toiminnallisuuksien osalta. Suunnittelijat suosivat interaktiivisten prototyyppien käyttöä, koska ne mahdollistavat kattavien ja tarkkojen käyttäjätestauksien pitämisen arviointivaiheessa. Käyttäjällä on testattavana prototyyppi, joka vastaa hyvin tarkasti kehitettävää ohjelmistoa. Käyttäjät testaavat prototyyppiä käymällä lävitse suunnittelijoiden ennalta määrittelemiä tehtäviä tai he pääsevät vapaasti kokeilemaan prototyyppejä ja kertomaan mielipiteensä siitä. Interaktiivisen prototyypin avulla myös asiakas saa selvemmän käsityksen siitä, mitä he ovat tilaamassa. Prototyyppi helpottaa asiakasta ilmaisemaan mielipiteensä ja antamaan palautetta, kun he pystyvät näkemään ja testaamaan konkreettista ohjelmistoprototyyppejä. Lisäksi interaktiivisesta prototyypistä on apua kehittäjille, koska he voivat verrata koodattua ohjelmistoa prototyyppiin. Näin varmistetaan, että lopullinen ohjelmiston kehitysversio vastaa suunnitelmia. Interaktiiviset prototyypit tehdään erilaisilla suunnitteluohjelmilla, joista suunnittelijat mainitsivat Sketch- sekä InDesign-ohjelmien käytön. Seuraavassa kahdessa lainauksessa suunnittelijat kertovat käyttävänsä interaktiivisia prototyyppejä, koska ne mahdollistavat käytettävyydestestaukset ja tätä kautta varmistamisen siitä, että projekti on menossa oikeaan suuntaan:

”Usein just klikattavia prototyyppejä näytetään käyttäjille ja sanotaan kokeile tällä tehä suorittaa tämmönen homma, että onnistuuko.” Suunnittelija 7

”Et ne (käyttäjät ja asiakas) näkee ihan visuaalisesti sen miltä se tulee näyttään ja mistä sä klikkaat ja miten tää toimii, se ehkä sitten niinkun tavallaan tuo semmosta varmistusta siihen, et ollaanks menty oikeeseen suuntaan.” Suunnittelija 8

Suunnitteluohjelmilla tehtäviä interaktiivisia prototyyppejä perusteltiin sillä, että niillä prototyyppien tekeminen on nopeaa ja edullista. Nopeasti tehtävät prototyypit sopivat ketterään kehitykseen hyvin, koska palautetta tulee kerätä nopeasti nopeissa iteraatioissa. Seuraavassa lainauksessa suunnittelija kertoo interaktiivisen prototyypin tekemisen olevan nopeaa ja niillä voidaan nopeasti kerätä palautetta:

”Se ei oo kovin monen päivän juttu yleensä (interaktiivinen prototyyppi), että päästään nopeesti hakeen se palaute asiakkaalta ja loppukäyttäjiltä.” Suunnittelija 4

Vaikka suunnittelijat näkevät prototyypit oleellisena osana suunnittelua niidenkin käytössä on havaittu rajoitteita. Osa haastatelluista suunnittelijoista kertoi, että etenkin pienemmissä projekteissa ei tehdä prototyyppejä ollenkaan, vaan ohjelmiston koodaaminen aloitetaan ensimmäisien tehtyjen näyttökuvien avulla. Seuraavassa lainauksessa suunnittelija kertoo, kuinka kehittäminen aloitetaan pelkästään käyttöliittymäkuvien avulla:

”Toisinaan tehdään vaan käyttöliittymäkuvia, mitkä ei oo interaktiivisa, et päästään liikkeelle tai aletaan koodaan saman tien niitä.” Suunnittelija 4

Ongelmana prototyypin käytössä kerrottiin niiden liian pitkälle tekeminen. Jos prototyyppi tehdään aikaisessa vaiheessa jopa liian valmiiksi, niin suunnitelmien iterointi kärsii. Tämä sotii ketterää käyttäjäkeskeistä suunnittelua vastaan, koska suunnitelmia tulisi iteroida. Seuraavassa kahdessa lainauksessa suunnittelijat kertovat, kuinka he haluavat välttää prototyypin liian pitkälle viemistä, johon helposti päädytään, koska muuten ollaan palattu vesiputousmallin mukaiseen ohjelmistokehitykseen:

”Haluaisin päästä niistä enemmän eroon, koska prototyyppien tekeminenkin on aika paljon sitä big design up frontia se johtaa helposti siihen, että tehdään suunnitelmat todella valmiiksi ennenkö kehitetään yhtään mitään.” Suunnittelija 2

”Se ydin kunhan toimii (prototyypin) ja se pistetään samantien käyttäjille ja testataan ja sit taas korjataan, että muuten ollaan taas perinteisessä vesiputouksessa, jos lähetään tosiaan tekeen maailma valmiiksi.” Suunnittelija 6

Prototyypin käyttäminen on tärkeässä osassa käyttäjäkeskeistä suunnittelua haastattelutulosten perusteella, mutta niissä on omat riskinsä. Kirjallisuuskatsauksen tutkimukset [Anwar et al., 2014, s. 162-163; Brhel et al., 2015, s. 175; Silva da Silva et al., 2011, s. 82] painottivat myös prototyypin käyttämistä, mutta eivät nostaneet yhtään niiden käytön mahdollisia haittoja esille.

5.1.6 Sidosryhmät

Viimeisenä pääteemana on sidosryhmät. Haastatellut suunnittelijat kertoivat loppukäyttäjän sekä asiakkaan rooleista osana integroitua ohjelmistokehitystä. Alateemoiksi muodustuikin: loppukäyttäjän rooli ja asiakkaan rooli.

Loppukäyttäjän rooli selvisi haastatteluista ensimmäiseksi alateemaksi sidosryhmien pääteemaan. Suunnittelijat kertoivat, että loppukäyttäjät pyrittiin pitämään mukana koko ohjelmistoprojektin ajan. Projektit lähtevät loppukäyttäjistä liikkeelle ja loppukäyttäjään palataan aina projektin eri vaiheissa uudestaan. Näin estetään väärin olettamuksien syntyminen. Alla olevassa lainauksessa suunnittelija kertoo, että loppukäyttäjä tulee pitää mukana projekteissa, jotta ei tehdä vääriä olettamuksia:

”Tosi tärkeä, koska niille se tehdään (loppukäyttäjä) eikä meille itsellemme tai sille asiakkaalle, että omat mielipiteet ja oletukset saattaa olla vääriä.” Suunnittelija 7

Haastattelutulokset ovat linjassa kirjallisuuskatsauksen kanssa. Chamberlain et al. [2006, s. 151] tekemän tutkimuksen perusteella käyttäjä tulisi myös pitää mukana koko kehitysprosessin ajan. Käyttäjäkeskeinen suunnittelu vaatii suoran yhteyden ohjelmiston loppukäyttäjään. Jotta ohjelmisto saavuttaa sille asetetut vaatimukset onkin jo aikaisessa vaiheessa pidettävä käyttäjä mukana. [Brhel et al., 2015, s. 173-174]

Loppukäyttäjät pidetään projekteissa mukana havainnointien, haastattelujen ja käytettävyydestien avulla. Käytettävyydestit nähdään erityisesti tärkeinä, koska loppukäyttäjä on ohjelmiston lopullinen käyttäjä, joten hänen tulee osata käyttää ohjelmistoa jo käytettävyydestauksissa. Osa haastateltavista mainitsi kuitenkin pelkästään käytettävyydestien käyttämisen osana käyttäjän osallistamista projektiin ilman syväluotaavia haastatteluja. Seuraavassa kahdessa lainauksessa kerrotaan, kuinka käyttäjää haastatellaan alun tutkimusvaiheessa ja osallistutetaan uudestaan, kun jotain konkreettista on luotuna kuten prototyyppejä:

”Oot haastatellu ymmärrysvaiheessa, niin niiden osallistaminen sit taas kun on jotain konkretiaa (hahmotelmat, prototyypit).” Suunnittelija 6

”Pyritään tottakai päästä haastatteleen tai näkeen niitä loppukäyttäjiä, mistä saadaan sitten se varmistus ja tota sitten tosiaan projektin aikana koitetaan sitten sitä, mitä oltiin tekemässä testata (prototyypit).” Suunnittelija 4

Brhel et al. [2015, s. 173] tekemän kirjallisuuskatsauksen perusteella käyttäjien osallistaminen käytettävyydesteihin on suosituin tapa osallistuttaa loppukäyttäjiä käyttäjäkeskeisessä ketterässä ympäristössä. Tämä on samassa linjassa haastattelutulosten kanssa.

Suunnittelijat haluavat pitää loppukäyttäjät mukana projekteissa, mutta loppukäyttäjien saaminen on pitkälti kiinni asiakkaasta. Projekteissa pyritään pitämään mukana useaa eri loppukäyttäjää, jolla voidaan varmistaa uusien näkökulmien saaminen. Haastatellut suunnittelijat kuitenkin kertoivat, että osalta loppukäyttäjien edustajilta mielipiteet voivat helposti jäädä kysymättä, ellei suunnittelija aktiivisesti pyydä palautetta. Seuraavassa lainauksessa suunnittelija kertoo, kuinka loppukäyttäjiltä voi jäädä mielipiteet kysymättä, jollei itse ole aktiivinen:

”Kyllä se usein niin menee, että monelta jää mielipide kysymättä, ellei me sitä aktiivisesti haeta.” Suunnittelija 5

Loppukäyttäjiä hankitaan yleensä projektikohtaisesti asiakkaan kautta, mutta osa suunnittelijoista kertoi hyödyntävänsä vanhoja käyttäjäkontakteja. Jos loppukäyttäjiä ei ole saatavilla projektitiimin muut jäsenet suorittivat käytettävyydestauksia. Yksi

haastatelluista suunnittelijoista kertoi, että loppukäyttäjien hankkiminen oli täysin ulkoistettu toiselle yritykselle. Tämä mahdollisti käyttäjien saamisen aina tarvittaessa. Kirjallisuuskatsauksen tutkimuksessa havaittiin, että käytettävyydestien järjestämisen varmistaminen vaatiikin etukäteissuunnittelua käyttäjien osallistumisen suhteen, käyttäen hyödyksi vuokrayrityksiä, tuttuja käyttäjiä tai etäkäytettävyydestejä [Salah et al., 2014, s. 6-7].

Asiakkaan rooli selvisi viimeiseksi alateemaksi sidosryhmistä puhuttaessa. Asiakkaan rooli luonnollisesti projekteissa on erityisesti päättää, mitä ominaisuuksia kehitetään. Suunnittelijat kertoivat, että päätökset suunnitteluratkaisuista tehdään yleensä asiakkaan-edustajan ja product ownerin kanssa. Product ownerin roolia keskusteluissa painotettiin, koska hän yleensä tietää parhaiten liiketoiminnan kannalta tärkeimmät asiat. Seuraavassa lainauksessa kuvataan, kuinka asiakas ja product owner vastaavat päätöksienteosta:

”Käytännössä asiakas ja meidän sisäinen product owner keskustelee keskenään. Product owner on henkilö, jolla on varmaan paras ymmärrys siitä mikä on asiakkaan liiketoiminnan kannalta tärkein asia tehdä.” Suunnittelija 5

Osa suunnittelijoista kertoi, että päätöksientekemiseen voi päästä vaikuttamaan myös kehitystiimin suunnittelijat ja kehittäjät yhdessä asiakkaan kanssa. Tällöin erityisesti suunnittelija pääsee kertomaan, mitä voidaan tehdä missäkin ajassa ja budjetin puitteissa. Alla oleva lainaus kertoo, kuinka päätökset tehdään yhdessä asiakkaan, suunnittelijan ja kehittäjän kanssa:

”Siinä käydään sellasta vuoropuhelua ehkä just sen kanssa, että ninku on suunnittelija, toteuttaja ja on asiakas ja neuvotellaan siitä, että mitä kannattais mistäkin näkökulmasta ja tota minkä verran mikäkin maksais.” V7

Haastatellut suunnittelijat kertoivat, että asiakas tulee osallistuttua projektiin heti alusta alkaen, jotta asiakas ei kohtaa ikäviä yllätyksiä projektin edetessä. Projektit elävät paljon suunnittelun ja kehityksen edetessä, tällöin voidaan joutua tekemään ratkaisuja, jotka eivät aina vastaa asiakkaan alkuperäistä näkemystä. Tämän takia asiakkaan tulee olla mukana tiiviisti projektissa, jotta hän on tietoinen kaikista ratkaisuista ja voi hyväksyttää tarvittavat muutokset. Asiakas voi myös arvioida suunnittelijoiden tuotoksia ja antaa palautetta. Seuraavassa lainauksessa suunnittelija kertoo, että asiakas tulee pitää mukana projektissa alusta alkaen:

”Joo se on äärimmäisen tärkeää että saadaan se osaaminen näkemys ja sitouttaminen siinä kohtaan. Kun lopputulos saattaa olla jotain muuta kun se ihanne.” Suunnittelija 5

Asiakkaan rooli voi olla hyvin näkyvä, koska haastatellut suunnittelijat kertoivat kehitystiimien välillä olevan samassa tilassa asiakkaan luona. Tällöin kehitystiimillä oli aina tarvittaessa käytettävissä joku henkilö asiakkaalta. Seuraavassa kahdessa lainauksessa suunnittelijat kertovat, kuinka kehitystiimi voi työskennellä asiakkaan tiloissa yhdessä asiakkaan edustajan kanssa:

”Yleensä meidän tiimit istuu asiakkaan luona eli optimaalisesti siinä on asiakkaalta joku henkilö joka vastaa projektista.” Suunnittelija 7

”Jonkin verran on-site projekteja eli ollaan asiakkaan luonakin on sitten niinkun tyyppejä töissä tai itseasiassa saattaa mennä kokonainen tiimi meiltä asiakkaan tiloihin tekemään töitä.” Suunnittelija 6

Kirjallisuuskatsauksen tutkimukset puolsivat asiakkaan aktiivista osallistamista ohjelmistoprojekteihin. Aivan kuten haastattelutuloksissa todettiin myös Chamberlain et al. [2006, s. 151] selvittivät tutkimuksessaan, että käyttäjän lisäksi asiakkaan tulee olla aktiivinen tiimin jäsen eikä vain passiivinen sivullinen.

5.2 Tyypit

Teemoittelu analyysissä syntyneiden teemojen pohjalta saatiin koottua kaksi integrointimallityyppiä, joilla käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integraatiota voidaan toteuttaa ohjelmistoprojekteissa. Ensimmäinen tyyppi painottaa suunnittelua ja toinen on tasapainotettu. Malleista voidaan valita jompikumpi sen mukaan, onko projektissa tarkoitus panostaa suunnitteluun vai tasapainoiseen ohjelmistokehitykseen.

5.2.1 Suunnittelupainotteinen integraatiomalli

Suunnittelupainotteisessa integraatiomallissa käyttäjäkeskeisen suunnittelun ja ketterän kehityksen prosessin tulee nojautua ketteryyteen ja nopeasti takaisin palaamiseen, jos suunnittelussa huomataan ongelmia tai korjattavaa. Suunnittelijan tulee olla itse aktiivinen suunnitelmien esille tuomisessa, jotta käyttäjäkeskeisyys ei huku ketteryyteen. Prosessin suunnittelu- ja kehityspolut suunnittelupainotteisessa mallissa tulisi nähdä yhtenä polkuna, joka kulkee yhdessä kiedottuna yhteen. Suunnitelmien tulee pysyä iteraation kehitystoimien edellä ja ohjelmointitöitä ei saa aloittaa ennen ensimmäisien suunnitelmien valmistusta, jotta vältytään turhalta kehitystyöltä. Koko integraatioprosessin kokonais kuvan hahmottamiseksi suunnittelupainotteisessa mallissa tulisi kokonaiskuva pitää mukana suunnittelijan omissa suunnitelmissa, kommunikoimalla aktiivisesti ja projektinhallintatyökalulla (esim. Jira).

Ketteristä menetelmistä suunnittelupainotteisuudessa tulisi suosia sovellettua Scrum-menetelmää. Scrumiin tulisi sekoittaa muita ketteriä menetelmiä, kuten Lean-ajattelua, jotta

ketterä kehitys saadaan sopimaan käyttäjäkeskeisyyden kanssa yhteen. Suunnittelijoita kehitystiimissä tulisi olla ainakin kaksi, jotta he voivat tukea toisiaan suunnitteluissa ja tehdä parisuunnittelua. Useampi suunnittelija toimii myös vakuutuksena projektille, jos toinen estyy kesken projektin, niin projektin teko ei pysähdy tästä. Suunnittelijan pitäisi osallistua suurimpaan osaan kaikista kehitystiimitapaamisista, vaikka asiat eivät suunnittelijaa aina koskisikaan, koska juuri tällä tavalla voidaan löytää uusia mahdollisuuksia suunnittelulle. Lisäksi aktiivinen osallistuminen tapaamisiin pitää suunnittelijan paremmin mukana projektin kulussa. Suunnittelijoiden välillä roolit tulisi jakaa käyttäjäkoke-mussuunnittelijaan, interaktiosuunnittelijaan ja käyttöliittymäsuunnittelijaan. Näin suunnittelijat voivat keskittyä omiin rooleihinsa ja välttää suunnittelijan ylityöllistämistä.

Etukäteen suunnittelussa esivaiheessa tulee alussa tavata asiakas, jonka kanssa käydään lävitse projektin vaatimukset ja tavoitteet. Loppukäyttäjiä tulisi nähdä heti esivaiheessa ja kartuttaa ymmärrys heistä havainnointien sekä haastattelujen avulla. Tärkeintä on välttää väärin olettamuksien tekeminen loppukäyttäjien tarpeista. Kehitystoimia (ohjelmointi) ei tule aloittaa vielä esivaiheen suunnitteluvaiheessa. Varsinaisen etukäteissuunnittelun alkaessa esivaiheen jälkeen tulee suunnittelussa keskittyä prototyyppeihin ja käytettävyydesteihin. Kun suunnitellut tuotokset käyvät lävitse kehitysvaihetta, suunnittelijat tekevät laadunvarmistusta. He seuraavat aktiivisesti kehitykseen tilaa ja antavat palautetta kehitetyistä ominaisuuksista. Jos kehitetty ominaisuus ei vastaa uusimpia suunnitelmia, se annetaan kehittäjälle uudestaan työstettäväksi. Kehitettävien ominaisuuksien valmistuessa suunnittelijat suorittavat niille käytettävyydestauksia itse ja loppukäyttäjillä. Laadunvarmistuksen takaamiseksi tulisi käyttää Kanban-tauluja, jotka sisältävät tarkastusvaiheet jokaiselle kehitettävälle ominaisuudelle. Suunnittelutoimia suhteuttaessa aikaan on tärkeää, että suunnittelija osallistuu projektien aloituspalaveriin, joissa hän voi itse kertoa kuinka paljon suunnittelutoimia ehdittää tekemään missäkin aikarajoissa. Suunnitteluun käytettävää aikaa tulee korostaa suunnittelupainotteisessa lähestymistavassa. Samoin rahan suhdetta suunnittelutoimiin tulee korostaa. Suunnittelulle pitää varata tarpeeksi rahoitusta etenkin käyttäjä tutkimusvaiheeseen, jotta kattava ymmärrys käyttäjistä saadaan muodostettua ja projektille saadaan luotua varma pohja. Suunnittelijan omilla näkemyksillä tulee olla suuri painoarvo keskusteltaessa projektin suunnitteluresursseista.

Suunnitteluratkaisuista kommunikoitaessa suunnittelupainotteisessa mallissa kehitystiimillä tulee olla yhteiset työskentelytilat. Tämä helpottaa ajatustenvaihtoa sekä vähentää muodollisten tapaamisten tarvetta, koska kommunikointia tapahtuu valmiiksi yhteisissä työtiloissa. Kehitystiimi tulisi sijoittaa asiakkaan tiloihin, jos mahdollista, koska tällöin asiakkaan edustaja on aina paikalla ja mahdollistaa suunnitteluideoista kommunikoinnin myös asiakkaalle. Lisäksi yhteinen työskentelytila kasvattaa luottamusta

suunnittelijoiden ja kehittäjien välillä. Yhteiset työskentelytilat eivät kuitenkaan takaa, että kommunikointia kehittäjien ja suunnittelijoiden välillä tapahtuu. Tämän takia dialogia tulee ylläpitää aktiivisesti ja oma-aloitteisesti henkilökemioiden ylitse. Kehittäjien ei myöskään pidä tehdä olettamuksia, että suunnitelmat tiedetään tarpeeksi hyvin, jolloin suunnittelu ja kommunikointi katkeavat. Kommunikointiin liittyy menetelmiä ja välineitä, joista prototyyppien käyttö painottuu kommunikointivälineenä. Konkreettiset prototyypit suunnittelijan selityksien lisäksi pystyvät kertomaan suunnitteluratkaisut sekä suunnitteluvision kehittäjille. Kommunikoinnissa viestintäsovelluksen (esim. Slack) käyttäminen nopeuttaa ja tukee aktiivista kommunikointia viestittelyn muodossa. Näiden lisäksi kehittäjät tulisi osallistuttaa tapaamisiin ensimmäisistä suunnitelmista, jotta saadaan varmistus siitä, mitä pystytään kehittämään.

Suunnittelupainotteisessa mallissa tutkimusvaiheen toimintatapoihin ja välineisiin kuuluu haastattelut, havainnoinnit ja työpajat. Näiden avulla luodaan kattava ymmärrys loppukäyttäjistä. Suunnittelu- ja arviointivaiheessa puolestaan korostuu persoonat, konseptit, rautalankamallit, prototyypit ja käytettävyydestaukset. Ensimmäiset suunnitelmat luodaan persoonien, konseptien ja rautalankojen avulla, joilla tuodaan ohjelmiston rakenne ja toiminnallisuudet esille. Tämän jälkeen prototyyppien käyttäminen painottuu. Prototyyppien tulee olla interaktiivisia, joista käy ilmi ohjelmiston toiminnallisuudet, ulkoasu ja kokonaisvaltainen käyttökokemus. Prototyyppien pääasiallisena tehtävänä on toimia välineinä käytettävyydestauksissa, joita suoritetaan loppukäyttäjillä. Niillä myös havainnollistetaan kehittäjille kehitettävää ohjelmistoa. Prototyyppien käytössä tulee kuitenkin muistaa niiden iterointia, eikä niitä tule viedä suunnittelun alussa liian pitkälle, jottei ajauduta takaisin vesiputousmalliin.

Sidosryhmistä loppukäyttäjä ja asiakas tulee pitää mukana koko ohjelmistoprojektin ajan. Projekti lähtee käyttäjästä ja loppuu käyttäjään. Näin varmistetaan, että projektin edetessä ei tehdä vääriä olettamuksia. Loppukäyttäjää pidetään mukana havainnointien, haastattelujen ja käytettävyydestien avulla. Käytettävyydestit painottuvat suunnittelupainotteisessa mallissa, koska loppukäyttäjä on kuitenkin ohjelmiston lopullinen käyttäjä, joten hänen tulee myös osata käyttää ohjelmistoa ja sen tulee tarjota paras mahdollinen käyttökokemus. Lisäksi suunnittelijan tulee aktiivisesti hakea käyttäjäpalautetta, jotta kaikilta käyttäjän edustajilta saadaan kysyttyä mielipide. Käytettävyydestit tulee tehdä loppukäyttäjillä eikä esimerkiksi projektitiimin jäsenillä. Asiakkaan tai asiakkaan edustajan tulee olla näkyvässä roolissa ohjelmistokehityksen aikana, eikä vain passiivisena sivullisena. Asiakkaan tulee olla tietoinen kaikista projektiin kohdistuneista muutoksista, jotta nämä eivät tule yllätyksenä projektin lopussa. Kehitystiimin tulisikin työskennellä asiakkaan tiloissa, jolloin asiakas on aina tarvittaessa käytettävissä, ja hän voi antaa palautetta projektin edetessä.

5.2.2 Tasapainotettu integraatiomalli

Tasapainotetussa integraatiomallissa käyttäjäkeskeisen suunnittelun ja ketterän kehityksen prosessin tulee painottaa myös ketteryyttä suunnittelussa. Suunnittelu- ja kehityspotut etenevät samana polkuna, mutta kehityksen ohjelmointityöt (back end) voidaan aloittaa jo suunnittelun alkuvaiheessa, jotta kaikki resurssit saadaan heti käyttöön. Tasapainotetussa mallissa kokonaiskuvan havainnointiin voidaan katsoa riittävän vain projektinhalintatyökalu (esim. Jira).

Ketterä menetelmä tasapainotetussa mallissa tulisi valita kehitystiimin ja asiakkaan kesken siten, että se tukee suunnittelua ja kehitystä tasapuolisesti. Suunnittelijoita kehitystiimissä voi olla vain yksi tasapainotetun ja resurssien säästämiseksi. Samanlainen tasapainotettu kuuluu myös suunnittelijan osallistumiseen kehitystiimintapaamisiin. Suunnittelijan ei tule osallistua kaikkiin kehitystiimin tapaamisiin, jotta hän voi keskittyä suunnittelutöihin. Suunnittelija osallistuukin vain suunnittelutapaamisiin ja kommunikoi muuten kehittäjien kanssa projektin edetessä. Suunnittelija tulee nähdä moniosaajana usean roolin kanssa. Yksi suunnittelija tekee käyttäjäkokemussuunnittelua, interaktiosuunnittelua ja käyttöliittymäsuunnittelua. Tasapainotuksen saavuttamiseksi suunnittelija osallistuu suunnittelutöiden lisäksi myyntityöhön ja ohjelmointiin (front end).

Etukäteen suunnittelun esivaiheessa aluksi tavataan asiakas, jonka kanssa käydään lävitse projektin vaatimukset ja tavoitteet. Loppukäyttäjää tulisi nähdä esivaiheessa vähintään haastattelujen muodossa, mutta mielellään myös havainnointia tulisi tehdä alun käyttäjätutkimuksessa, jotta voidaan muodostaa kattava ymmärrys ohjelmiston lopullisista käyttäjistä. Tasapainotetussa mallissa voidaan aloittaa ohjelmointityöt jo esivaiheessa, jotta kaikki resurssit saadaan käyttöön. Esivaihesuunnittelun jälkeen jatkuvassa etukäteissuunnittelussa suunnittelutyöt vähenevät, koska suurin osa suunnittelusta tulee toteuttaa esivaiheessa. Suunnittelua jatketaan kuitenkin prototyyppinä iteroimalla. Suunnittelijat tekevät laadunvarmistusta kehitetyille ominaisuuksille seuraamalla kehitystä ja antamalla korjausehdotuksia resurssien mukaan. Suunnittelutoimia suhteuttaessa aikaan ja rahaan on tärkeää, että suunnittelija pääsee osallistumaan projektien aloituspalaveriin, joissa hän voi itse kertoa kuinka paljon suunnittelutoimia ehdittää tekemään missäkin aikarajoissa. Suunnittelun ei kuitenkaan pidä korostua tasapainotetussa mallissa, vaan muillekin kehitysvaiheille tulee varata tasapuolisesti resursseja.

Tasapainotetussa mallissa suunnitteluratkaisuista kommunikoinnin ei tarvitse tapahtua yhteisissä työtiloissa, vaikka tätä suositetaan, jos se on mahdollista. Kehitystiimi voi olla maantieteellisesti hajautettu, jolloin kommunikointi koko kehitystiimin kesken tapahtuu pääasiassa etätapaamisilla. Kommunikointia tulee käydä aktiivisesti suunnittelijoiden ja kehittäjien välillä suunnitteluratkaisuista, jotta vääriä olettamuksia ei tehdä

suunnitelmista. Prototyyppi on tärkeä väline suunnitteluratkaisuista kommunikoitaessa, koska näiden avulla kehittäjät pääsevät helposti käsiksi suunnitteluratkaisuihin. Tarpeen mukaan tasapainotetussa mallissa voidaan käyttää viestittelysovelluksia kommunikoinnin tukena.

Tasapainotetussa mallissa käytetään toimintatapoja ja välineitä, mitä resurssien puitteissa pystyy. Ymmärryksen luomiseen voidaan valita menetelmät haastatteluista, havainnoinneista ja työpajoista. Pääpaino tulee antaa kuitenkin haastatteluille, koska näiden avulla voidaan saada jo tarvittava syvä tieto loppukäyttäjistä. Suunnitteluvaiheessa voidaan tehdä käyttöliittymäkuvia tai prototyyppejä. Tasapainotetussa mallissa voidaan kehitystyö aloittaa jo käyttöliittymäkuvien pohjalta. Arviointivaiheessa prototyypille tai suoraan ohjelmoidulle ohjelmistolle tulee tehdä käytettävyystestaus.

Sidosryhmistä loppukäyttäjä ja asiakas tulee pitää mukana ohjelmistoprojektin tietyissä vaiheissa. Loppukäyttäjiä pidetään mukana haastattelujen, havainnointien ja käytettävyydestien avulla resurssien puitteissa. Painotus tulee pitää haastatteluissa ja käytettävyydestauksissa, koska loppukäyttäjän tulee osata käyttää lopullista ohjelmistoversiota. Loppukäyttäjien ollessa vaikeasti saatavissa tai resurssipulan takia heitä ei voi käyttää, voi projektitiimin jäsenet suorittaa käytettävyydestauksia. Asiakas tulee pitää myös mukana ohjelmistoprojektissa niin paljon kuin mahdollista. Asiakkaalle tulee vähintään ilmoittaa kaikki muutokset projektin edetessä, jotta projektin lopussa asiakkaalle ei tule yllätyksenä tehdyt muutokset.

5.2.3 Integraatiomallien yhteenveto

Tuloksena syntyi siis 2 eri integraatiomallia. Taulukossa 11 on yhteenveto tyypittelyn integraatiomalleista.

Taulukko 11. Tyypittely teemoittelun teemojen pohjalta.

Käyttäjakeskeisen suunnittelun ja ketterän kehityksen integrointimallit		
	Painotus suunnittelussa	Tasapainotettu
Integroitu prosessi		
Prosessin piirteet	Ketteryys ja nopea takaisin palaaminen suunnitelmissa, jos ne vaativat muutoksia. Suunnittelijan aktiivisuus.	Ketteryys ja nopea takaisin palaaminen suunnitelmissa, jos ne vaativat muutoksia.
Suunnittelu- ja kehityspotut	Yhdessä samana polkuna eteneminen. Suunnittelu aina edellä kehitystä.	Yhdessä samana polkuna eteneminen, mutta kehitys voi mennä myös suunnittelun edelle.
Kokonaiskuvan hahmottaminen	Kokonaiskuva suunnitelmissa, kommunikoinnissa ja projektinhallintatyökalussa.	Vähintään projektityökalun käyttäminen.
Suunnittelija osana kehitystiimiä		
Ketterämenetelmä	Sovellettu Scrum.	Menetelmän valinta tiimin ja resurssien mukaan.
Suunnittelijoiden määrä	2-3 suunnittelijaa.	1 suunnittelija.
Tiimitapaamiset	Osallistuminen kaikkiin kehitystiimin tapaamisiin.	Vain suunnittelutapaamiset.
Suunnittelijan roolit	Käyttäjäkokeilusuunnittelu, käyttöliittymäsuunnittelu ja interaktiosuunnittelu.	Moniosaaja suunnittelija.
Etukäteen suunnittelu		
Esivaihesuunnittelu	Tarpeiden määrittely sekä ymmärryksen muodostaminen käyttäjästä. Ohjelmointia ei aloiteta.	Ymmärryksen muodostaminen käyttäjästä. Back end -ohjelmoinnin voi aloittaa samalla.
Etukäteissuunnittelu	Painotus prototyyppien iteroinnissa ja käytettävyydesteissä. Suunnittelu yhden iteraation kehityksen edellä.	Prototyyppien iterointia, mutta suunnittelutyöt vähenevät.
Suunnittelun laadunvarmistus	Kehityksen aktiivinen seuraaminen, käytettävyydestaukset ja Kanban.	Kehityksen seuraaminen.
Suunnittelu suhteessa aikaan	Suunnittelijan osallistaminen aloituspalaveriin.	Suunnittelijan osallistaminen aloituspalaveriin.
Suunnittelu suhteessa rahaan	Suunnittelijan osallistaminen aloituspalaveriin. Rahoituksen korostaminen suunnittelussa.	Suunnittelijan osallistaminen aloituspalaveriin.
Kommunikointi suunnitteluratkaisuista		
Työskentelytila	Kehitystiimi samassa tilassa.	Kehitystiimi hajautettu tai samassa tilassa.
Dialogin käyminen	Aktiivinen ja jatkuva kommunikointi henkilökemioiden yli.	Kommunikointi vähintään uusista suunnitteluratkaisuista.
Menetelmät ja välineet kommunikoinnissa	Kommunikointi prototyypeillä, viestittelysovelluksella ja suunnittelutapaamisilla.	Prototyypeillä kommunikointi.
Toimintatavat ja välineet suunnittelussa		
Tutkimusvaiheen toimintatavat ja välineet	Haastattelut, havainnoinnit, työpajat.	Haastattelut.
Suunnittelu- ja arviointivaiheen toimintatavat ja välineet	Persoonat, konseptit, rautalankamallit, prototypit ja käytettävyydestaukset.	Käyttöliittymäkuvat tai prototypit.
Sidosryhmät		
Loppukäyttäjä	Osana kehitystiimiä.	Vaihteleva osallistuminen.
Asiakas	Osana kehitystiimiä.	Vaihteleva osallistuminen.

6. Pohdinta ja johtopäätökset

Tässä tutkimuksessa selvitettiin, kuinka käyttäjäkeskeinen suunnittelu ja ketterä kehitys voitaisiin integroida suunnittelijan näkökulmasta. Tutkimuksella haluttiin saada selville, millä käytänteillä ja menetelmillä nämä kaksi eri kehitysmenetelmää voidaan yhdistää onnistuneesti. Seuraavaksi esitetään tutkimuksen keskeiset tulokset linkitettynä aiempiin tutkimuksiin. Tässä luvussa luodaan katsaus myös keskeisiin johtopäätöksiin, jatkotutkimusmahdollisuuksiin ja tämän tutkimusprosessin arviointiin.

6.1 Tutkimustulokset ja kirjallisuuskatsaus

Käyttäjäkeskeisen suunnittelun ja ketterän kehityksen yhdistämisessä haastatellut suunnittelijat painottivat erityisesti ketteryyden tuomaa nopean suunnittelun ja takaisin palautuksen mahdollisuutta, jos suunnitelmiin tarvitsee tehdä muutoksia. Käyttäjäkeskeisen suunnittelun katsottiin sopivan hyvin yhteen tämän ketterän periaatteen kanssa. Ketteryyden nopeuden nostivat esiin myös Jurca et al. [2014, s. 25] ja Brhel et al. [2015, s. 171-172] tutkimuksissaan käyttäjäkeskeistä suunnittelua hyödyntävänä tekijänä, koska monet nopeat iteraatiot mahdollistavat usean ohjelmistoarvioinnin ja tätä kautta palautteen mukaisten korjausehdotusten nopean tekemisen ohjelmistoon. Ketteryyden näkemisen hyvänä osana suunnittelua voi vaikuttaa se, että käyttäjäkeskeinen suunnittelu ja ketterä kehitys kummatkin perustuvat iteratiiviseen kehittämiseen. Kummatkin menetelmät tukevat toisiaan. Chamberlain et al. [2006, s. 144] kertoivat tutkimuksessaan suunnittelun ja ketteryyden samankaltaisuuksista, joista ensimmäisenä he kertoivat kummankin olevan iteratiivisia prosesseja ja näin tukevan toisiaan.

Haastatellut suunnittelijat kertoivat käyttäjäkeskeisen suunnittelun ja ketterän kehityksen yhdistämisen tuovan esille suunnittelijoiden ja kehittäjien omista näkemyksistä kiinni pitämisen ja toistensa kyseenalaistamisen, jotka puolestaan voivat luoda ongelmia ohjelmistokehitykselle. Mielenmalleissa suunnittelijat ajattelevat käyttäjän näkökulmasta, mutta kehittäjät puolestaan voivat ajatella, että ohjelmistoa kehitetään suunnittelijoille, koska he eivät ole tekemisissä loppukäyttäjien kanssa. Haastatteluissa kerrottiin, kuinka suunnittelijoiden tulee olla näkyvästi läsnä, koska kehittäjät tulee saada luottamaan ja ymmärtämään suunnittelua ja sen tärkeyttä osana ohjelmistokehitystä. Kirjallisuuskatsauksen tutkimukset toivat esille myös eroja suunnittelijoiden ja kehittäjien välillä. Integroidussa ympäristössä suunnittelijoiden ja kehittäjien välillä voi olla tiettyä valtakamppailua, koska suunnittelu voidaan nähdä enemmän toisarvoisempana kehitykseen verrattuna [Chamberlain et al., 2006; Jurca et al., 2014; Salah et al., 2014]. Tämä tietynlainen suunnittelu vastaan kehitys voi johtua ketterän kehityksen piirteistä. Toimiva ohjelmisto toimii ketterässä kehityksessä ensisijaisena edistymisen mittarina [Agile Alliance, 2001]

ja suunnittelu ei. Suunnitelmat eivät tuota konkreettista ohjelmistoa toisin kuin ohjelmointityö, mikä voi saada suunnittelun näyttämään toisarvoiselta kehitykseen verrattuna.

Suunnittelu- ja kehityspolut nähtiin etenevän yhdessä haastattelujen perusteella. Suunnittelijat painottivat, että suunnittelua ja kehitystä ei pitäisi erottaa, ja kuinka niiden tulisi kulkea yhdessä, jotta pystytään varmistamaan ohjelmiston laatu. Alussa kuitenkin on yksi iteraatio omistettu pääasiassa suunnittelulle, mutta haastatellut suunnittelijat kertoivat, että tässä vaiheessa kehittäjät voivat jo alkaa tekemään back end -ohjelmointia. Tämä suunnittelu- ja kehityspolkujen yhdessä kulkeminen on myös linjassa Fox et al. [2008], Najafi ja Toyoshiba [2008], Silva da Silva et al. [2011] ja Sy [2007] tutkimuksien kanssa, vaikka he painottivat, että mitään ohjelmointitöitä ei tulisi aloittaa ennen kuin ensimmäinen suunnitteluiteraatio on valmistunut, jonka jälkeen kehitys ja suunnittelu alkavat vasta kulkemaan yhdessä. Back end -ohjelmoinnin aloittaminen aikaisessa vaiheessa voi selittyä sillä, että kehittäjien puoli halutaan osallistuttaa työntekoon heti projektin alussa, jotta kaikki resurssit saadaan käyttöön. Lisäksi tämän totesi yksi haastatelluista suunnittelijoista. Integroidussa ympäristössä tulisikin huolehtia siitä, että ensimmäiselle suunnitteluiteraatiolle varataan aikaa ennen ohjelmointitöiden aloittamista.

Ketteristä menetelmistä käyttäjäkeskeisen suunnittelun kanssa haastatellut suunnittelijat kertoivat käyttävänsä Scrum-menetelmää, johon oli liitetty muita ketteriä menetelmiä ja näiden ominaisuuksia. Scrumiin liitettyihin muihin ketteriin menetelmiin kuului Lean-ajattelu ja Kanban. Kirjallisuuskatsauksen tutkimukset puolsivat myös Scrum-menetelmän käyttämistä [Anwar et al., 2014; Brhel et al., 2015; Chamberlain et al., 2006; Najafi ja Toyoshiba, 2008; Salah et al., 2014; Silva da Silva et al., 2013]. Scrum kuuluukin yhä tunnetuimpiin ja käytetyimpiin ketteriin menetelmiin. Yhdysvaltalaisen VersionOne [2018, s. 9] yrityksen teettämän kyselyn mukaan 56 % vastaajista käytti Scrum-menetelmää. Myös Rodríguez et al. [2012] teettämän tutkimuksen mukaan Scrum-menetelmä kuului käytetyimpiin. Haastateltavat suunnittelijat eivät kuitenkaan kertoneet käyttävänsä puhdasta Scrum-menetelmää vaan tietynlaista hybridiversiota. Scrum luo viitekehyksen kehitykselle, mutta se tarvitsee tuekseen muita ketteriä menetelmiä ja näiden periaatteita, jotta käyttäjäkeskeisyys saadaan sopimaan yhteen ketteryyden kanssa. Puhtaassa ketteryydessä painotetaan muun muassa toimivaa ohjelmistoa ennen kattavaa dokumentointia [Agile Alliance, 2001]. Suunnittelu on tietynlaista dokumentointia, jota tarvitaan käyttäjäkeskeisessä suunnittelussa, joten ketteryyttä ei voida orjallisesti yhdistää käyttäjäkeskeiseen suunnitteluun.

Kehitystiimissä haastatellut suunnittelijat suosivat kahden suunnittelijan käyttämistä. Tätä perusteltiin sillä, että näin pystyttiin tekemään parisuunnittelua, jakamaan rooleja ja jo projektin vakuutuksen kannalta oli parempi käyttää kahta suunnittelijaa.

Kirjallisuuskatsauksen tutkimukset tukivat myös kahden suunnittelijan käyttämistä, mutta hieman eri syystä. Tutkimuksissa [Salah et al., 2014, s. 7; Williams ja Ferguson, 2007, s. 4] suositettiin kahta suunnittelijaa suuren suunnittelutyötaakan takia. Jurca et al. [2014] kertoivat, kuinka suunnittelijat eivät ehtineet tekemään kaikkia suunnittelutoimia valmiiksi työmäärien takia. Haastatellut suunnittelijat eivät kuitenkaan kertoneet valtavista työtaakoista, joiden takia olisi tarvittu useampaa suunnittelijaa, vaan painottivat kahden suunnittelijan yhteistyön tärkeyttä ja roolien jakamista ohjelmistokehityksessä.

Tiimitapaamisien osalta osa suunnittelijoista kertoi käyttävänsä Scrumin tarjoamia päivä- ja viikkotapaamisia projektista riippuen. Haastatellut suunnittelijat kertoivat, kuinka suunnittelijat eivät osallistu kaikkiin kehitystiimin tapaamisiin, koska heidän työnsä eivät etene niin nopeasti, että niistä pitäisi raportoida usein ja kommunikointi yhteisissä tiloissa vähentää tapaamisien määrää. Tiimitapaamisista jättäytymistä pois pidettiin myös haasteellisenä, koska tällöin voitiin tippua projektista tai menettää mahdollisia suunnittelumahdollisuuksia. Kirjallisuuskatsauksen tutkimuksessa [Salah et al., 2014, s. 6] puolestaan kerrottiin, kuinka tärkeää suunnittelijan osallistaminen päivittäisiin tapaamisiin kehittäjien kanssa oli, koska tämä mahdollisti työaktiviteettien synkronoinnin ja suunnitelmien selventämisen. Tapaamisiin osallistumisessa tulisikin löytää sopiva tasapaino, koska työajan käyttäminen esimerkiksi kehityspainotteisiin tapaamisiin voi viedä suunnittelijan arvokasta työskentelyaikaa.

Suunnittelijoilla voi olla monia erilaisia rooleja haastattelujen mukaan. Suunnittelijan rooleihin voi kuulua käyttäjäkokemussuunnittelu, käyttöliittymäsuunnittelu, interaktiosuunnittelu, myynnillinen- ja liiketoiminnallinentuki ja front end -ohjelmointi. Haastatteluissa kerrottiin useaan otteeseen roolista, jossa suunnittelija teki käyttäjäkokemussuunnittelua, interaktiosuunnittelua ja käyttöliittymäsuunnittelua. Suunnittelijaa pidetään moniosajana, koska hänellä on usein monta eri roolia. Tässä voi painottua myös haastattelutuloksista suunnittelijoiden omat mielipiteet. Lisäksi kirjallisuuskatsauksen tutkimukset kertoivat, kuinka suunnittelijalla on monia rooleja. Silva da Silva et al. [2013] tekemässä tutkimuksessa havaittiin ketterän kehityksen ja käyttäjäkeskeisen suunnittelun integraatiossa muodostuneen kolme eri suunnittelijan roolia: käyttäjäkokemussuunnittelija, interaktiosuunnittelija ja käyttöliittymäsuunnittelija. Kyseiset roolit ovat havaittavissa myös haastattelutuloksista. Jurca et al. [2014, s. 28-29] puolestaan selvittivät tutkimuksessaan, että suunnittelijan rooleihin voi sisältyä suunnittelun lisäksi liiketoiminnallista tukea sekä ohjelmointia. Nämäkin suunnittelijan roolit löytyivät haastattelutuloksista. Sekä kirjallisuuskatsauksen tulokset että haastattelutulokset kummatkin puoltavat sitä, että suunnittelija nähdään moniosajana ja usean roolin omaavana. Osittain tätä monen roolin omaamista voidaan varmaan selittää sillä, että käyttäjäkeskeinen suunnittelu itsessään sisältää useita eri suunnittelutoimia. On myös varmasti luonnollista yhdistää suunnittelijan

työtehtäviin muutakin kuin suunnittelua, kuten myyntiä ja ohjelmointia. Myyntivaiheissa suunnittelijat voivat itse kertoa, mitä voidaan suunnitella missä ajassa ja budjetissa. Suunnittelijalla tulisi olla vahvaa myyntiosaamista, jotta hän voi perustella suunnittelutoimet ja vakuuttaa asiakas ostamaan suunnittelutyön [Bruun et al., 2018, s. 11]. Lisäksi omien suunnitelmien konkretisointi itse ohjelmoimalla varmistaa sen, että suunnitelmat implementoidaan oikein.

Suunnittelutoimia tehdään etukäteen ennen varsinaista käyttöliittymän ja toimintojen ohjelmointia. Suunnittelijat kertoivat, että etukäteen suunnittelu alkaa esivaihesuunnittelulla, jossa ensin määritellään projektin tarpeet asiakkaan kanssa sekä selvitetään loppukäyttäjien tarpeet tarkasti. Esivaiheessa kehittäjät alkavat myös tekemään jo back end -ohjelmointia. Esivaihesuunnittelu oli suurilta osin samassa linjassa kirjallisuuskatsauksen kanssa, koska nämäkin painottivat ymmärryksen muodostamisen tärkeyttä esivaiheessa [Brhel et al., 2015; Chamberlain et al., 2006; Najafi ja Toyoshiba, 2008]. Tutkimukset eivät kuitenkaan suosineet minkään ohjelmointityön aloittamista vielä esivaihesuunnitteluvaiheessa, vaan pelkästään suunnitteluun tulisi keskittyä [Silva da Silva et al., 2011; Brhel et al., 2015]. Kehittäjien ohjelmointityön aloittamista voi selittää se, että halutaan välttyä tehottomuudelta kehittäjien puolella projektin alussa. Brhel et al. [2015] suosittelivat kuitenkin kehittäjien osallistamista suunnittelutoimiin esivaiheessa, eikä ohjelmointitöihin. Haastatteluissa kerrottiin, että esivaiheen jälkeen suunnittelua jatketaan etukäteissuunnittelulla, jotta suunnitelmat pysyvät aina kehitystöiden edellä. Myös Fox et al. [2008], Najafi ja Toyoshiba [2008], Silva da Silva et al. [2011] ja Sy [2007] kertoivat tutkimuksissaan suunnittelun kulkevan yhden iteraation kehitystä edellä. Tämä onkin luonnollinen tapa jatkaa ohjelmistokehitystä, jotta kehittäjät voivat implementoida uudet suunnitelmat aina ajallaan.

Suunnittelujen laadunvarmistuksessa suunnittelijat eivät nähneet tiukkoja standardeja, vaan laadunvarmistus oli suunnittelijan ja kehittäjän omasta aktiivisuudesta kiinni. Aiemmissä tutkimuksissa [Silva da Silva et al., 2011, s. 83-84; Fox et al., 2008, s. 67-68] puolestaan kerrottiin, kuinka kehittäjien tuotoksia pitäisi tarkistaa aina tietyin väliajoin iteraatioissa, jotta voidaan varmistaa kehittäjien noudattaneen suunnitelmia. Yksi haastateltavista kuitenkin kertoi Scrumin kanssa Kanbanin käytöstä osana laadunvarmistusta, koska tämä sisältää oman tarkastusvaiheen. Tämä on hyvä esimerkki siitä, kuinka Scrumiin voidaan sekoittaa muita ketteriä menetelmiä, kuten Lean ajattelun mukaista Kanbania. Leanin periaatteisiin kuuluukin hukan (Lean-käsitteet on selitetty alakohdassa 2.2.2) poistaminen jatkuvalla laadunvarmistuksella [Poppendieck ja Poppendieck, 2006].

Haastatellut suunnittelijat kertoivat, että heillä on tarpeeksi aikaa suorittaa suunnittelutoimet, koska he yleensä voivat osallistua asiakastapaamisiin ja kertoa, mitä voidaan tehdä

missäkin ajassa. Kaksi suunnittelijaa kuitenkin mainitsivat ajan vähäisyyden suunnittelu-toimien tekemisessä. Jurca et al. [2014] ja Salah et al. [2014] kertoivat tutkimuksissaan, kuinka suunnittelijoilla ei ole tarpeeksi aikaa tehdä kaikkia heille osoitettuja suunnittelu-toimia erityisesti useiden roolien ja projektien takia. Haastattelutuloksissa kerrottu Suun-nittelutoimista suhteessa rahoitukseen suunnittelijat kertoivat, että he pystyvät tekemään kaiken, mitä on suunniteltu, mutta haluaisivat tehdä enemmän. Rajallinen budjetti rajaa suunnitteluaktiviteetteja. Alun käyttäjätutkimukseen tulisi varata enemmän budjettia, jotta pystytään luomaan kattava ymmärrys loppukäyttäjistä jo ohjelmistoprojektin alussa. Yllättävää oli, että kirjallisuuskatsauksen tutkimukset eivät nostaneet rahoitusta ja bud-jettia esille osana suunnittelua. Salah et al. [2014, s. 5-6] kertoivat käytettävyydestien nostavan budjettia likaa, mikä nähtiin ongelmallisena, mutta muuten kirjallisuuskatsauk-sen tutkimukset eivät kertoneet rahan vaikutuksesta suunnitteluun, vaikka haastattelutu-losten perusteella tämä nousi keskeiseksi. Tähän voi vaikuttaa se, että kirjallisuuskatsauk-sen tutkimuksista suurin osa nojautui tutkijoiden omiin havaintoihin ja päätelmiin, eikä suunnittelijoiden mielipiteisiin, mutta tämäkään ei selitä, miten rahan ja budjetin suhde on voitu sivuuttaa. Tutkimuksista osa myös sisälsi suunnittelijoiden haastatteluja. Työ-kulttuurilliset tekijät voivat olla osa syy, miksi ulkomaalaiset tutkimukset, eivät käsitel-leet rahoituksen suhdetta suunnittelutoimiin. Suomalaisessa työ kulttuurissa eriarvoisuutta pidetään huonona asiana ja työntekijöiden suhde esimiehiin ei ole etäinen [Ylöstalo, 2007, 18], joten Suomessa budjetista ja rahoituksesta osana suunnittelua on helpompi pu-hua vapautuneemmassa työympäristössä.

Keskeisenä haastattelutuloksena oli suunnitteluratkaisuista kommunikoinnin tärkeys. Kaikki suunnittelijat kertoivat yhteisen työskentelytilan olevan oleellisessa asemassa suunnitteluratkaisujen kommunikoinnin kannalta. Myös aiemmat tutkimukset [Brhel et al., 2015; Jurca et al., 2014; Williams ja Ferguson, 2007] kannattivat yhteisiä tiloja, koska tämä mahdollisti paremman yhteistyön ja kommunikoinnin suunnittelijoiden ja kehittä-jien välillä. Työskentelytilan lisäksi dialogia tulee käydä aktiivisesti, koska yhteinen työ-skentelytila ei tätä aina takaa haastattelujen perusteella. Kommunikoinnin nähtiin painot-tuvan eniten projektien alkuun ja loppuun, jonka takia aktiivista kommunikointia tuleekin ylläpitää jatkuvasti, jotta suunnitteluideat ymmärretään ja implementoidaan oikein. Va-jaan dialogin käymisestä kertoivat Najafin ja Toyoshiban [2008] tapaustudkimuksessaan, kun kommunikointi suunnittelijoiden ja kehittäjien välillä hiipui, implementoitiin suun-nitteluratkaisut väärin ja kaikki tuleva työ jouduttiin käyttämään jo luotujen ominaisuuksien korjaamiseen. Kommunikointikatkokset voivat selittyä sillä, että suunnitelmat olete-taan tietävän jo tarpeeksi hyvin ja suunnittelijat sekä kehittäjät uppoutuvat omiin töihinsä. Suunnittelijat liittivät kommunikointiin tiettyjä välineitä ja menetelmiä. Näistä mainittiin piirrosten ja prototyyppien käyttäminen kommunikoinnin apuna, koska näillä pystyttiin

havainnollistamaan suunnitteluideoita kehittäjille, loppukäyttäjille ja asiakkaalle. Salah et al. [2014] kertoivat tutkimuksessaan myös, kuinka suunnitteluratkaisut voidaan jakaa kehittäjille suunnitteluvälineissä, kuten prototyyppien muodossa. Lisäksi haastatellut suunnittelijat kertoivat viestittelysovelluksien käyttämisen ja kehittäjien osallistamisen suunnittelutapaamisiin hyvinä ratkaisuinä kommunikaation ylläpitämisessä. Erinäisten viestittelysovellusten käyttämistä ei kuitenkaan mainittu muissa tutkimuksissa, vaikka usea haastateltu suunnittelija kertoi näiden tärkeydestä osana aktiivista kommunikointia.

Haastatellut suunnittelijat kuvasivat yleisimpiä toimintatapoja ja välineitä osana suunnittelua käyttäjäkeskeisessä ketterässä ympäristössä (suunnittelukäsitteet selitetty alakohdassa 2.3.3). Ymmärryksen muodostamiseen suunnittelijat käyttivät työpajoja, haastatteluja ja havainnointeja. Anwar et al. [2014] ja Brhel et al. [2015] kertovat myös tutkimuksessaan erityisesti tilannetutkimuksien (havainnointi ja haastattelut) olevan käytetyimpiä menetelmiä integroidussa ympäristössä. Tämä on linjassa haastattelutulosten kanssa. Tilannetutkimukset kuuluvatkin käyttäjäkeskeisen suunnittelun keskeisiin menetelmiin ja niiden avulla saadaankin kattava ymmärrys käyttäjistä käyttäjän käyttöympäristössä (Taulukko 4).

Kaikki haastatellut suunnittelijat kertoivat käyttävänsä prototyypejä jossain määrin osana suunnittelua, koska näillä voidaan konkretisoida suunnitteluideat kehittäjille, käyttäjille ja asiakkaalle. Niillä pystytään hahmottamaan koko ohjelmisto toimintoihin ilman ohjelmointityötä. Kirjallisuuskatsauksen tutkimuksista myös Brhel et al. [2015], Jurca et al. [2014] ja Anwar et al. [2014] kertoivat prototyyppien kuuluvan käytetyimpiin välineisiin osana suunnittelua, koska ne tarjoavat tarpeeksi tarkat kuvaukset kehittäjille kehitettävistä toiminnallisuuksista. Prototyyppien painotusta voidaan perustella sillä, että ne kuuluvat käyttäjäkeskeisen suunnittelun 12 päätavoitteeseen [Gulliksen et al., 2003, s. 401-403]. Ensimmäisinä prototyypeinä toimii matalantason prototypit, kuten rautalangat, joilla hahmotellaan enemmän toiminnallisuutta. Näitä seuraa korkeantason interaktiiviset prototypit, joissa painottuu ohjelmiston ulkoasu ja kokonaisvaltainen käyttäjäkokemus. Prototypit ovat keskeisessä osassa käytettävyydestä, joita tehdään loppukäyttäjillä. Haastattelutulokset prototyyppien osalta on linjassa kirjallisuuskatsauksen tutkimuksien [Brhel et al., 2015, s. 175; Silva da Silva et al., 2011, s. 82] kanssa, koska näissä kerrottiin integroidussa mallissa prototyyppien alkavan matalan tason prototyypeistä, joilla kuvataan toiminnallisuudet, joita seuraa aina korkean tason prototypit. Prototyyppien suosiota selittää niiden tarjoamat monet hyödyt. Ne ovat edullisia, nopeita, havainnollistavia ja mahdollistavat käytettävyydestä. Haastatteluissa mainittiin kuitenkin prototyyppien tuomat mahdolliset ongelmat, joita kirjallisuuskatsauksen tutkimuksissa ei mainittu ollenkaan. Suunnittelijat kertoivat, että prototyyppien tekemisessä on vaarana niiden liian pitkälle vieminen, jolloin niiden iterointi unohtuu ja ollaan palattu

takaisin vesiputousmalliseen työstämiseen. Hieman kummallista olikin, että muut tutkimukset eivät nostaneet prototyypin mahdollisia ongelmia esille, ottaen huomioon niiden tärkein roolin osana käyttäjäkeskeistä suunnittelua. Tähän voi vaikuttaa se, että kirjallisuuskatsauksen tutkimukset perustuivat suurilta osin tutkijoiden omiin havaintoihin, eikä suunnittelijoiden näkemyksiin prototyypin käyttämisestä. Niissä myös korostettiin suunnittelun tärkeyttä nostamatta esiin mahdollisia ongelmia tai haittoja suunnittelutoimissa.

Sidosryhmistä haastatellut suunnittelijat kertoivat loppukäyttäjän ja asiakkaan rooleista osana ketterää käyttäjäkeskeistä suunnittelua. Suunnittelijat kertoivat, että loppukäyttäjä pyritään pitämään mukana koko projektin ajan, jotta vältetään väärin olettamuksen tekeminen. Ymmärryksen muodostamisen jälkeen haastatteluilla ja havainnoilla luodaan konseptit ja prototyypit, joita testataan käyttäjillä. Chamberlain et al. [2006, s. 151] tutkimuksen perusteella käyttäjä tulisi myös pitää mukana koko kehitysprosessin ajan. Samoin Brhel et al. [2015, s. 173] tutkimuksen kirjallisuuskatsauksessa he selvittivät käyttäjien osallistamisen käytettävyydestä olleen suosituin tapa osallistuttaa loppukäyttäjiä käyttäjäkeskeisessä ketterässä ympäristössä. Käyttäjän osallistamista käytettävyydestä hin painotetaan varmaan sen takia, koska näin saadaan parhaiten varmuus siitä, että lopullinen käyttäjä todella osaa käyttää ohjelmistoa ja, että se luo parhaan mahdollisen käyttökokemuksen. Käyttäjäkeskeisen suunnittelun periaatteisiin katsoessa fokus tuleekin pitää käyttäjässä ja käyttäjän tarpeet ohjaavat kehitystä [Gulliksen et al., 2003, s. 401-403].

Sidosryhmistä asiakkaan osallistamista haastatellut suunnittelijat pitivät tärkeänä. Asiakkaan rooli painottuu, koska hän viimekädessä päättää kehitettävän ohjelmiston ominaisuuksista. Asiakas tulee sitouttaa projektiin ja hänen näkemyksensä tulee olla saatavilla, jotta asiakas tietää koko ajan projektin tilanteen ja välttyy yllätyksiltä suunnitteluratkaisuissa. Optimaalisena vaihtoehtona pidettiin, että kehitystiimi on sijoitettuna asiakkaan luokse. Chamberlain et al. [2006, s. 151] kertoivat tutkimuksessaan myös, että käyttäjän lisäksi asiakkaan tulee olla aktiivinen tiimin jäsen eikä vain passiivinen sivullinen integroidussa käyttäjäkeskeisessä ketterässä ympäristössä.

6.2 Keskeiset johtopäätökset

Tämän tutkimuksen tuloksilla voidaan edesauttaa käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integroimista toisiinsa. Tutkimuksessa selvisi integraatiota vahvistavia tekijöitä, hyväksi koettuja käytäntöjä sekä ongelmia, joita tulisi välttää. Tutkimuksen perusteella tein seuraavat keskeiset johtopäätökset käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integroinnin osalta:

Integroitu prosessi

- Ketteryteen kuuluvan nopeuden ja takaisin palaamisen painottaminen suunnittelussa.
- Suunnittelu tulee nähdä yhtä tärkeänä kehityksen kanssa, jotta vältetään valtakamppailulta suunnittelijan ja kehittäjän välillä sekä luotua luottamusta suunnittelua kohtaan.
- Suunnittelu- ja kehityspolkujen tulee edetä yhdessä tukien toisiaan.
- Integroidun prosessin kokonaiskuvan mielessä pitäminen kommunikaation ja omien suunnitelmien avulla.

Suunnittelija ja kehitystiimi

- Sovelletun Scrum-menetelmän käyttäminen, johon on lisätty muiden ketterien menetelmien ominaisuuksia, kuten Lean-ajattelua. Ketteryyden tulee toimia yhteen suunnitteluperiaatteiden kanssa, joten puhdasta yhtä ketterää menetelmää ei voida ottaa orjallisesti käyttöön.
- Suunnittelijan osallistaminen suurimpaan osaan kehitystiimin tapaamisista, jotta suunnitelmat saadaan pidettyä esillä ja suunnittelija ei tipu projektista. Kehittäjiä koskeviin tapaamisiin osallistuminen vain, jos suunnittelijalla aikaa, koska näissä voidaan löytää uusia suunnittelutarpeita.
- Kahden suunnittelijan käyttäminen parisuunnittelun ja ohjelmistoprojektin vakuutuksen takia.
- Suunnittelijan roolien jakaminen käyttäjäkokemussuunnittelijaan, interaktiivisuunnittelijaan ja käyttöliittymäsuunnittelijaan. Mahdollisuuksien mukaan suunnittelija voidaan osallistuttaa myös myynnintyöhön ja front end -ohjelmointiin.

Etukäteen suunnitteleminen

- Esivaihesuunnittelussa painotus ymmärryksen muodostamiseen loppukäyttäjistä haastatteluilla ja havainnoinnilla (tilannetutkimus). Kehittäjät eivät saa aloittaa ohjelmointitöitä vielä, mutta voivat osallistua suunnittelutöihin.
- Etukäteissuunnittelun tekemistä yksi iteraatio kehityksen edellä, jossa painottuu prototyypit ja käytettävyydestaukset.
- Suunnittelijoiden tulee tehdä laadunvarmistusta seuraamalla aktiivisesti kehitettyjen ominaisuuksien tilaa. Kehitettyjä ominaisuuksia verrataan suunnitelmiin ja niille tehdään käytettävyydestauksia.
- Suunnittelija tulee osallistuttaa projektin alun asiakastapaamisiin, joissa suunnittelijat voivat kertoa, mitä suunnittelutoimia ehditään tekemään aikarajoissa.

- Projektin rahoituksesta keskusteltaessa tulee painottaa alun käyttäjätutkimuksen tärkeyttä, jotta budjettia ei kuluteta myöhemmin turhien ominaisuuksien kehittämisellä.

Kommunikointi suunnitteluratkaisuista

- Yhteinen työskentelytila suunnittelijoilla ja kehittäjillä edesauttaa kommunikointia.
- Aktiivista dialogia tulee käydä koko ohjelmistoprojektin ajan, eikä kommunikointi saa painottua olettamuksien takia vain projektin alkuun ja loppuun, jotta suunnitteluratkaisuja ei implementoida väärin.
- Prototyyppejä tulee käyttää suunnitteluratkaisujen ja suunnitteluvision kommunikoinniksi.

Yleiset toimintatavat ja välineet suunnittelussa

- Tutkimusvaiheessa ymmärryksen muodostamiseen käyttäjästä käytetään haastatteluja ja havainnointia (tilannetutkimus) sekä työpajoja.
- Suunnittelu- ja arviointivaiheessa käytetään persoonia, konsepteja, rautalankamalleja, prototyyppejä ja käytettävyydestestauksia.
- Interaktiivisten prototyyppien käyttöä tulee painottaa suunnittelussa. Interaktiivisia prototyyppejä käytetään havainnoimaan ohjelmistoa kokonaisuudessaan ulkoasusta kokonaisvaltaiseen käyttäjäkokemukseen.

Sidosryhmät

- Loppukäyttäjä tulee osallistuttaa koko projektin ajaksi projektiin haastatteluun, havainnointiin ja käytettävyydestestien avulla.
- Asiakas ei saa olla passiivinen sivustakatsoja vaan aktiivinen kehitystiimin jäsen. Kehitystiimin työskentelemistä asiakkaan tiloissa suositetaan.

6.3 Jatkotutkimusaiheet

Tämä tutkimus käsitteli käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integraatiota suunnittelijan näkökulmasta, joten uusien näkökulmien esille tuominen voisi antaa lisää ratkaisuja onnistuneen integraation saavuttamiselle. Tutkimuksen tuloksiin on vaikuttanut suunnittelijoiden omat näkemykset ja mielipiteet, joita kehittäjät ja muut sidosryhmät voivat kokea eri tavalla. Haastattelutuloksissa muun muassa suunnittelijat kokivat asemansa hieman heikommaksi verrattuna kehittäjiin. Suunnittelijat kertoivat, että heiltä vaaditaan enemmän aktiivisuutta verrattuna kehittäjiin. Tulokset voisivat olla erilaiset, jos kehittäjiltä kysyttäisiin mielipidettä käyttäjäkeskeisen suunnittelun ja ketterän

kehityksen yhdistämisestä. Lisäksi loppukäyttäjien ja asiakkaiden haastatteleminen sekä tutkiminen vaikuttaa hyvältä tavalta saada uutta tietoa integraation onnistumiselle. Koskeehan käyttäjakeskeinen suunnittelu nimenomaan loppukäyttäjää ja vastaavanlaisia tutkimuksia ei ole aikaisemmin tehty.

Mielenkiintoisena jatkotutkimusaiheena voisi olla myös tässä tutkimuksessa kahden luodun integrointimallin (suunnittelupainotteinen- ja tasapainotettu malli) toteuttaminen käytännössä ja tutkiminen. Kummatkin mallit voitaisiin toteuttaa projektikohtaisesti riipuen, onko projektin pääpaino suunnittelussa vai ei. Tutkija voisi tehdä havaintoja projektin kulusta ja haastatella projektiin osallistuvia sidosryhmiä. Tuloksena voisi syntyä vielä pidemmälle jalostetummat integraatiomallit.

Käyttäjakeskeisessä ketterässä ympäristössä sekä haastatellut suunnittelijat että kirjallisuuskatsauksen tulokset kertoivat prototyyppien käyttämisen tärkeydestä. Prototyyppien käytön hyödyt nostettiin esille erityisesti kirjallisuuskatsauksen tutkimuksissa, mutta niiden mahdollisista ongelmista ei puhuttu yhtään toisin kuin haastattelutuloksissa. Suunnittelijat kertoivatkin pitävänsä prototyypejä tärkeänä osana suunnittelua, mutta pelkäsivät niiden liian pitkälle viemisen haittaavan suunnittelua, koska tällöin suunnitelmia ei enää pystytty iteroimaan. Prototyyppien tutkiminen osana käyttäjakeskeisen suunnittelun ja ketterän kehityksen integraatiota vaikuttaa myös hyvältä jatkotutkimuskohteelta. Toisena keskeisenä erona kirjallisuuskatsaukseen havaittiin suunnittelun suhde projektien rahoitukseen. Haastatellut suunnittelijat kertoivat, kuinka projektien rajallinen budjetti rajoittaa etenkin alun tutkimusvaiheen suunnittelutoimien tekemistä, mutta kirjallisuuskatsauksen tutkimuksissa tätä ei nostettu esille, vaikka haastattelujen perusteella tämä vaikutti oleelliselta onnistuneen suunnittelun kannalta. Budjetin ja rahoituksen suhdetta suunnitteluun käyttäjakeskeisessä ketterässä ympäristössä olisikin syytä tutkia tarkemmin.

6.4 Tämän tutkimusprosessin arviointi

Tutkielman tekeminen oli pitkä prosessi, mutta se eteni mielestäni sujuvasti. Tutkimusprosessin alussa aihe eli hieman ja muutti muotoaan, koska tutkimusongelmaa lähdettiin aluksi käsittelemään liian laajasti. Tutkimusongelman rajauksen jälkeen prosessi lähti nopeasti liikkeelle. Erityisen hyödylliseksi osoittautui tutkimussuunnitelman huolellinen tekeminen, joka tuki tutkimusprosessin etenemistä. Varsinainen tutkielman kirjoittaminen alkoi teoriapohjan sekä kirjallisuuskatsauksen kirjoittamisella. Kirjallisuuskatsausta varten muita aiheeseen liittyviä tutkimuksia löytyi aluksi runsaasti yliopiston tarjoamista tietokannoista. Tutkimukset eivät kuitenkaan olleet suunnittelijan näkökulmasta toisin kuin tämä tutkimus. Valittavien tutkimuksien kesken oli kuitenkin tehtävä rajausta, jossa auttoi valintakriteerilistan (Liite 2) käyttäminen. Ainoastaan laadulliset tutkimukset

tutkimuskysymyksen kannalta valittiin mukaan. Näin saatiin kerättyä luotettavimmat ja tutkimusongelman kannalta oleelliset tutkimukset. Koko tiedonkeruuprosessin aikana itselleni muodostui kattava näkemys ketterän kehityksen sekä käyttäjäkeskeisen suunnittelun integroimisesta ja tähän liittyvistä oleellisista käsitteistä.

Kirjallisuuskatsauksen pohjalta saatiin luotua teemahaastatteluiden aiheet sekä kysymykset. Haastateltavien etsiminen osoittautui hieman haastavaksi, koska monet yritykset eivät listaa työntekijöitään yrityksen kotisivuille. Tämän takia haastateltavia joutui etsimään pitkään ja usein ensin piti lähestyä yrityksen henkilöstövastaavaa, ennen kun mahdollisen suunnittelijan yhteystiedot saatiin. Haastatteluissa olisikin voinut käyttää lumipallo-otantaa. Lumipallo-otannassa tutkijalla on avainhenkilö, joka johdattaa hänet seuraavan informantin luokse, jolloin tutkija etenee informantista toiseen tutkimusaineistoa kerätäkseen [Saaranen-Kauppinen ja Puusniekka, 2006]. Tämä olisi voinut nopeuttaa haastateltavien keräämistä.

Ensimmäinen positiivinen yllätys tutkimusprosessin aikana ilmeni ensimmäisen kymmenen haastattelukutsun lähettämisen jälkeen. Myöntäviä vastauksia haastattelukutsuihin tuli seitsemän ensimmäisen viikon aikana, joista suurin osa päivän sisällä kutsujen lähettamisestä. Aihe selvästi kiinnosti suunnittelijoita ja he olivat innokkaita osallistumaan haastatteluihin. Pienenä ongelmana oli kuitenkin haastatteluajkojen sopiminen. Muutamalla suunnittelijalla oli paljon työkiireitä, jolloin haastatteluajat jouduttiin sopimaan jopa reilun kuukauden päähän. Tästä ei kuitenkaan muodostunut suurta ongelmaa tutkimusprosessille, koska ensimmäisiä haastatteluja päästiin pitämään jo viikon jälkeen kutsujen lähettamisestä ja tätä kautta litteroimaan sekä analysoimaan.

Itse haastattelujen pitämisissä ei ilmennyt suuria ongelmia ja ne onnistuivat hyvin. Osa haastatteluista venyi jopa tunnin pituisiksi ja monesti niitä seurasi vielä pitkät jälkipuheet aiheesta, joka puolestaan puhui aiheen kiinnostavuudesta. Haastattelutilanteissa haastattelukysymykset muuttuivat hieman ensimmäisien haastattelujen jälkeen, ja näitä olisikin voinut hioa vielä enemmän ennen ensimmäisiä haastatteluja. Kysymyksissä oli hieman toistoa, jonka takia muutama alkuperäinen kysymys poistettiin. Kysymyksistä pari vaati myös täsmentämistä ensimmäisissä haastatteluissa niiden laajuuden takia, jonka takia näitä rajattiin hieman.

Haastatteluiden pitämisen jälkeen huomasin aineiston laajuuden. Litteroinnissa ja etenkin analysoinnissa kesti enemmän, mitä olin suunnitellut. Monet eri teemat havaittiin aineiston eri kohdissa useasti, minkä takia analysointivaihe kesti pidempään. Aineistoon syventymiseen ja auki kirjoittamiseen olisi voinut alussa varata enemmän aikaa. Kaikesta huolimatta aineisto saatiin käsiteltyä kunnolla lävitse ja tulokset tehtyä näkyviksi.

Kirjallisuuskatsauksen vertaaminen haastattelutuloksiin oli mielekästä, koska selviä eroja sekä samankaltaisuuksia oli havaittavissa.

Analysointivaihe eli hieman prosessin aikana, koska aluksi analysointi oli tarkoitus tehdä vain teemoittelun avulla. Ensimmäisien analyysien jälkeen kävi kuitenkin selväksi, että tyypittely olisi viisain tapa jatkaa analyysiä vielä pidemmälle. Aineiston avulla pystyinkin tekemään kaksi eri integrointimallia, kuinka käyttäjäkeskeinen suunnittelu ja ketterä kehitys pystytään yhdistämään. Tämä tiivistä tutkimustulokset mielekkäästi konkreettisiin malleihin.

Jos jotain olisi voinut tehdä vielä toisin, niin aihetta olisi voinut rajata vielä enemmän. Käyttäjäkeskeisyyden osalta olisi voinut keskittyä vain johonkin tiettyyn osa-alueeseen, kuten pelkkään kommunikointiin suunnitteluratkaisuista tai prototyyppien käyttämiseen. Haastatellut suunnittelijat nostivat useaan otteeseen esille myös projektiriippuvuuden, joten monesti vastauksissa jouduttiin tekemään yleistyksiä tai rajaamaan vastaus projektin kokoon. Aiheen olisikin voinut sitoa vain tietyn kokoisiin projekteihin. Aiheen laajuudesta huolimatta saatiin muodostettua hyvä kokonaisvaltainen kuvaus aiheesta.

Vaikka tutkimusongelmassa riitti tutkittavaa, oli tutkielman tekemiselle kuitenkin varattu tarpeeksi aikaa. Tutkielman tekeminen oli antoisaa sekä mielekästä, koska aihe oli kiinnostava ja siitä syntyi selviä uusia tuloksia. Aiheen monipuolinen tutkiminen antoi itselleni hyvät tiedot siitä, kuinka käyttäjäkeskeinen suunnittelu sekä ketterä kehitys voidaan integroida keskenään erilaisissa ohjelmistoprojekteissa.

6.5 Tämän tutkimuksen luotettavuus

Luotettavuuskysymys on oleellinen erityisesti tieteellisissä teksteissä. Laadullisen tutkimuksen arviointi pelkistyy kysymykseksi tutkimusprosessin luotettavuudesta. Lähtökohdana on tutkijan avoin subjektiviteetti ja sen myöntäminen, että itse tutkija on tutkimuksen keskeinen tutkimusväline. Laadullisen tutkimuksen luotettavuutta voidaan arvioida uskottavuuden, siirrettävyyden, varmuuden ja vahvistettavuuden avulla. [Eskola ja Suoranta, 1996, s. 151-153]

Uskottavuus luotettavuuden yhtenä kriteerinä tarkoittaa sitä, että vastaako tutkijan käsitteellistykset ja tulkinta tutkittavien käsityksiä [Eskola ja Suoranta, 1996, s. 153]. Tämän tutkimuksen uskottavuutta pyrittiin vahvistamaan aluksi teoriataustaan huolellisesti tutustumalla ennen aineiston keräämistä. Tämä mahdollisti suunnittelijoiden haastatteluissa mahdollisuuden selventää ja tarkentaa suunnittelijoiden vastauksia sekä käyttää heidän kanssaan samoja käsitteitä ja näin välttää väärinkäsityksiä ja vääriä tulkintoja. Itse analyysivaiheessa pyrin olemaan avoin ja peilaamaan omia ennakko-oletuksia.

Mahdollisimman tarkan aineistoanalyysin saavuttamiseksi käytettiin koodausta, jossa yksi sana tai sanapari määritti käsiteltävän teeman. Lisäksi tämän tutkielman uskottavuutta tukee se, että tutkimustulokset esitetään perusteellisesti haastattelulainauksien ja kirjallisuuskatsauksen vertaamisen avulla. Näin aineisto tuodaan näkyväksi lukijalle, jolloin hän voi itse arvioida tulkintojen oikeellisuutta.

Siirrettävyydellä tarkoitetaan tulosten siirrettävyyttä toiseen kontekstiin, joillain tietyin ehdoin [Eskola ja Suoranta, 1996, s. 153]. Tämän tutkimuksen tulosten siirrettävyys on rajoitettua, mutta mahdollista. Siirrettävyys tuleekin tehdä tietyin määritellyin ehdoin, vaikkei yleistykset ole monimuotoisuuden takia yleensä mahdollisia laadullisissa tutkimuksissa. Tulosten yleistettävyyttä olisi voinut parantaa laajemmalla otannalla ja toinen tutkija olisi voinut analysoida aineiston.

Varmuus luotettavuuskriteerinä tarkoittaa mahdollisuuksien mukaan huomioon otettavia tutkimukseen ennustamattomasti vaikuttavia tekijöitä [Eskola ja Suoranta, 1996, s. 153]. Varmuutta tässä tutkimuksessa on pyritty lisäämään huolellisella teemahaastattelurungon suunnittelulla ja sen testaamisella sekä haastattelutekniikoita harjoittelemalla. Haastatteluympäristönä toimi suurimmaksi osaksi haastateltujen omien yritysten tilat tai vaihtoehtoisesti Tampereen yliopistolta varattiin rauhallinen neuvotteluhuone haastattelua varten. Nauhoituslaitteet testattiin aina ennen haastatteluita ja varmuuden vuoksi nauhoitettiin kahdella eri laitteella. Itse litterointivaihe tehtiin tarkasti. Myös omat ennakko-oletukset tutkittavasta ilmiöstä tiedostettiin ja pyrittiin siihen, etteivät ne vaikuta aineiston analysointiin ja tulkintaan.

Vahvistuvuudella tarkoitetaan sitä, että tulkinnat saavat tukea toisista vastaavista tutkimuksista [Eskola ja Suoranta, 1996, s. 153]. Vahvistuvuutta lisää tässä tutkimuksessa se, että tämän tutkimuksen laadullisen tutkimuksen tutkimustuloksia esiteltäessä niitä verrattiin aiempiin käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integraation tutkimuksiin. Suurimpana erona tämän tutkimuksen ja muiden tutkimuksien välillä oli se, että tämä tutkimus on rajattu suunnittelijan näkemykseen käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integraatiosta. Muut tutkimukset tutkivat aihetta yleisemmällä tasolla ja tutkijan omien havaintojen avulla. Tulkintoja on pyritty tukemaan mahdollisimman tarkasti vastaavilla tutkimuksilla tekstin, kuvien ja taulukkojen avulla. Vaikka tutkielmassa oli mukana vain kahdeksan haastatellun suunnittelijan näkemykset, selvisi niistä samankaltaisia tuloksia aiempien tutkimuksien kanssa. Myös eroja havaittiin, mutta suurin osa oli asioiden painotuksissa.

6.6 Tämän tutkimuksen rajoitteet

Tässäkin tutkimuksessa oli omat rajoitteensa. Aivan aluksi tulee huomioda, että tässä tutkimuksessa tehdyllä kirjallisuuskatsauksen tutkimuksilla oli laaja otanta vuodesta 2005 vuoteen 2018. Tämä laaja vuosiotanta oli kuitenkin tarpeen kattavan ja luotettavan aineiston keräämiseksi. Suurimmat rajoitteet kirjallisuuskatsauksen tutkimuksissa olivat niiden taustat ja toteuttamistavat verrattuna tämän tutkimuksen empiiriseen tutkimukseen. Tutkimuksista yksikään ei ollut suomalainen ja ne eivät painottaneet suunnittelijän näkökulmaa. Tutkimuksia etsittiin myös vain Tampereen yliopiston tarjoamista tietokannoista, joten joitain oleellisia tutkimuksia on voinut jäädä tutkimatta. Toisaalta Tampereen yliopiston tietokannat valittiin, koska jokin rajausta oli tehtävä kerättävän aineiston suhteen ja aineiston haluttiin tulevan luotettavista tietokannoista. Näistä rajoitteista huolimatta uskon, että saadut tulokset ovat kattavat ja laadukkaat hyvälle teoriapohjalle.

Suurimmat rajoitukset koskivat haastateltavia suunnittelijoita. Haastateltavia oli vain kahdeksan ja voidaan olettaa, että tämä haastateltavien määrä ei anna tarpeeksi luotettavaa kokonaiskuvaa käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integraatiosta. Haastateltavien taustat olivat erilaiset, vaikka yhteistä kaikille oli käyttäjäkeskeinen suunnittelu ketterässä ohjelmistokehitysympäristössä. Haastateltavat tulivat myös erikokoisista ohjelmistoyrityksistä. Kaikesta huolimatta tämän tutkimuksen tuloksia voidaan pitää luotettavina, koska haastatteluissa toistuivat samat teemat ja tuloksia voitiin linkittää kirjallisuuskatsauksen tuloksiin.

7. Yhteenveto

Ketterä kehitys ei aina yksin takaa ohjelmistoa, jolla on käytettävyydeltään hyvä käyttöliittymä. Tästä johtuen käyttäjäkeskeisen suunnittelun yhdistäminen ketterään kehitykseen on herättänyt laajalti kiinnostusta. [Fox et al., 2008, s. 63] Menetelmien yhdistäminen mahdollistaa laadukkaiden ohjelmistojen kehittämisen ketterästi, mutta jotta nämä kaksi eri ohjelmistokehitysmenetelmää voidaan yhdistää, tulee integroinnin käytänteet kuitenkin olla kunnossa. Tämän tutkimuksen tavoitteena olikin löytää menetelmät ja käytännöt käyttäjäkeskeisen suunnittelun ja ketterän kehityksen yhdistämiseen suunnittelijan näkökulmasta. Löydetty menetelmät ja käytännöt tuli kuvata myös siten, että niitä pystytään hyödyntämään käytännön integroinnissa erilaisissa ohjelmistoprojekteissa. Integroinnin menetelmien ja käytäntöjen löytämiseksi haastateltiin kahdeksaa suunnittelijaa, joiden työnkuvaan kuului käyttäjäkeskeinen suunnittelu. Lisäksi oman empiirisen tutkimuksen tuloksia verrattiin tehdyn kirjallisuuskatsauksen tuloksiin, jolla nostettiin esille samankaltaisuuksia ja eroja.

Oman tutkimuksen tulokset olivat suurilta osin samassa linjassa kirjallisuuskatsauksen tutkimuksien kanssa, mutta eroavaisuuksia myös löytyi. Suurimmat erot ilmenivät ketterän menetelmän käytössä ja rahan suhteesta suunnittelutoimiin. Kirjallisuuskatsauksen tutkimukset puolsivat ketteristä menetelmistä puhtaan Scrum-menetelmän käyttämistä, mutta haastattelutulokset puolestaan muokattua Scrumia, jotta menetelmien integrointi saadaan toimivaksi. Yllättävää oli myös, että kirjallisuuskatsauksen tutkimukset eivät nostaneet esiin rahoituksen vaikutusta tehtäviin suunnittelutoimiin. Haastatellut suunnittelijat painottivat, että monesti he haluaisivat tehdä enemmän käyttäjätutkimusta, jotta saadaan luotua tarpeeksi kattava ymmärrys loppukäyttäjistä, mutta rajallisen budjetin takia tämä ei ole mahdollista. Keskeiseksi tulokseksi muodostui kommunikoinnin tärkeys suunnitteluratkaisuista. Muihin tuloksiin integroinnin kannalta selvisi ketteryyden korostaminen, suunnittelun ja kehityksen tiivis yhdessä kulkeminen, suunnittelutoimien pysyminen yhden iteraation kehityksen edellä, kahden tai useamman suunnittelijan käyttäminen omine rooleineen, esivaiheen käyttäjätutkimusvaiheen painottaminen, prototyyppien korostaminen, loppukäyttäjän aktiivinen osallistaminen ja asiakkaan kanssa tiiviin yhteistyön ylläpitäminen.

Aiheena käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integrointi on laaja ja siitä voidaan tehdä jatkotutkimusta. Tässä tutkimuksessa tutkittiin integraatiota suunnittelijan näkökulmasta. Kehittäjän, loppukäyttäjän tai asiakkaan näkökulmat voisivat antaa aiheeseen uuden näkökulman. Lisäksi tässä tutkimuksessa luotuja integraatiomalleja voisi tutkia ja jalostaa pidemmälle. Aiheena käyttäjäkeskeisen suunnittelun ja ketterän kehityksen integroiminen antaa paljon mahdollisuuksia jatkotutkimuksille ja sen tutkiminen on yhä tarpeellista, jotta voidaan tehdä ketterästi laadukkaita ohjelmistoja.

8. Viiteluettelo

Abrahamsson, P., Salo, O. ja Ronkainen, J. 2002. Agile software development methods : review and analysis. VTT Publications 478, Espoo.

Agile Alliance. 2001. Manifesto for agile software development. Viitattu: 11.10.2018. Osoitteesta: <http://agilemanifesto.org/>

Albert, W. and Tullis, T. 2013. Measuring the user experience: collecting, analyzing, and presenting usability metrics. Elsevier, Amsterdam.

Anwar, S., Motla, Y, H., Siddiq, Y., Asghar, S., Hassan, M, S. and Khan, Z, I. 2014. User-centered design practices in scrum development process: A distinctive advantage? 17th IEEE International Multi Topic Conference: Collaborative and Sustainable Development of Technologies, IEEEINMIC 2014 - Proceedings, s. 161-166.

Bias, R, G. and Mayhew, D, J. 2005. Cost-Justifying Usability: An Update for the Internet Age, 2nd Edition. Elsevier Science & Technology.

Brhel, M., Meth, H., Maedche, A. and Werder, K. 2015. Exploring principles of user-centered agile software development: A literature review. Information and Software Technology 61 (2015), s. 163-181.

Bruun, A., Larusdottir, M., Nielsen, L., Nielsen, P.A. and Persson, J.S. 2018. The Role of UX Professionals in Agile Development: A Case Study From Industry. In Proc. of the 10th Nordic Conference of Human-Computer Interaction (NordiCHI ' 18). ACM Press. s. 1-12.

Chamberlain, S., Sharp, H. and Maiden, N. 2006. Towards a Framework for Integrating Agile Development and User-Centred Design. Extreme Programming and Agile Processes in Software Engineering: Proceedings of the 7th International Conference (XP 2006), Berlin : Springer, s. 143-153.

Dybå, T. and Dingsøyr, T. 2008. Empirical studies of agile software development: A systematic review. Information and software technology, 50(9-10), s. 833-859.

Eskola, J. ja Suoranta, J. 1996. Johdatus laadulliseen tutkimukseen. Rovaniemi: Lapin yliopistopaino.

Fox, D., Sillito, J. and Maurer, F. 2008. Agile Methods and User-Centered Design: How These Two Methodologies Are Being Successfully Integrated in Industry. Agile Conference, Toronto, 4.8-8.8, 2008, s. 63-72.

- Gross, J. and McInnis, K. 2003. Kanban made simple. New York: AMACOM.
- Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J. and Cajander, Å. 2003. Key principles for user-centered systems design. *Behaviour and Information Technology*, 22(6), s. 397-409.
- Heinilä, J., Strömberg, H., Leikas, J., Ikonen, V., Iivari, N., Jokela, T., Aikio, K.-P., Jounila, I., Hoonhout, J. and Leurs, N. 2005. User-Centered Design Guidelines for Methods and Tools. University of Oulu, Dept. of Information processing science.
- Highsmith, J. and Cockburn, A. 2001. Agile Software Development: The Business of Innovation. *Computer*, 34(9), s. 120-122.
- Hirsjärvi, S., Remes, P. ja Sajavaara, P. 2007. Tutki ja kirjoita. 13. painos, Helsinki: Tammi.
- Hussain, Z., Slany, W. and Holzinger, A. 2009. Current State of Agile User-Centered Design: A Survey. Springer-Verlag Berlin Heidelberg.
- Hyysalo, S. 2009. Käyttäjä tuotekehityksessä – Tieto, tutkimus, menetelmät. Helsinki: Taideteollinen korkeakoulu.
- Ikonen, M., Pirinen, E., Fagerholm, F., Kettunen, P. and Abrahamsson, P. 2011. On the impact of Kanban on software project work: An empirical case study investigation. Published in: I. Perseil, K. Breitman & R. Sterrit, *Engineering of Complex Computer Systems (ICECCS)*, 2011 16th IEEE International Conference (s. 305–314). Las Vegas: IEEE Computer Society.
- ISO 9241-210. 2010. Ergonomics of human–system interaction — Part 210: Human-centered design for interactive systems.
- Janes, A. 2015. A Guide to Lean Software Development in Action. 2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW).
- Jurca, G., Hellmann, T. and Maurer, F. 2014. Integrating Agile ja User-Centered Design A Systematic Mapping and Review of Evaluation and Validation Studies of Agile-UX. Agile Conference, Kissimmee, 28.7.-1.8., 2014, s. 24-32.
- Kujala, S. 2003. User involvement: A review of the benefits and challenges. *Behaviour & Information Technology*, 22(1), s. 1–16.

Kuutti, W. 2003. Käytettävyys, suunnittelu ja arviointi. Helsinki: Talentum.

Lean Enterprise Institute, Principles of Lean, What is Lean. 2015. Viitattu: 12.4.2018.
Osoitteesta: <https://www.lean.org/WhatsLean/Principles.cfm>

Miller, G. 2001. The Characteristics of Agile Software Processes. Proceedings of the 39th International Conference and Exhibition of Object-Oriented Languages and Systems, (TOOLS 39), Santa Barbara, CA, s. 385-387.

Najafi, M. and Toyoshiba, L. 2008. Two Case Studies of User Experience Design and Agile Development. Agile Conference, Toronto, 4.8.-8.8., 2008, s. 531–536.

Nielsen, Jakob. 1993. Usability Engineering. Boston: Academic Press cop. s. 358.

Ovaska, S., Aula, A. ja Majaranta, P. 2005. Käytettävyystutkimuksen menetelmät. Tampere: Tampereen yliopisto.

Poppendieck, M. and Poppendieck, T. 2003 Lean Software Development: An Agile Toolkit. Boston: Addison Wesley.

Poppendieck, M. and Poppendieck, T. 2006. Implementing Lean Software Development: From Concept to Cash. Boston: Addison Wesley.

Poppendieck, M. and Cusumano, M, A. 2012. Lean Software Development: A Tutorial. IEEE Software 29 (5): 26–32. <https://doi.org/10.1109/MS.2012.107>.

Pratt, A. and Nunes, J. 2012. Interactive Design: An Introduction to the Theory and Application of User-centered Design. Rockport Publishers.

Preece, J., Rogers, Y., and Sharp, H. 2002. Interaction Design: beyond human-computer interaction. New York: John Wiley & Son, Inc.

Rekha Devi, Kh., Sen, A.M., and Hemachandran, K. 2012. A working framework for the user-centered design approach and a survey of the available methods. International Journal of Scientific and Research Publications. 2(4).

Rodriguez, P., Markkula, J., Oivo, M. and Turula, K. 2012. Survey on agile and lean usage in Finnish software industry. International Symposium on Empirical Software Engineering and Measurement (ESEM'12), Lund, Sweden, s. 139-148.

Saaranen-Kauppinen, A. ja Puusniekka, A. 2006. KvaliMOTV - Menetelmäopetuksen tietovaranto. Tampere: Yhteiskuntatieteellinen tietoarkisto. Viitattu: 10.4.2019. Osoitteesta: <http://www.fsd.uta.fi/menetelmaopetus/>

Salah, D., Paige, R, F. and Cairns, P, A. 2014. A systematic literature review for agile development processes and user centered design integration. In 18th International Conference on Evaluation and Assessment in Software Engineering, EASE '14, London, England, United Kingdom, 5.13-5.14, 2014. s. 1-10.

Schwaber, K. and Sutherland, J. 2017. Scrum Guide. Viitattu: 20.11.2018. Osoitteesta: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>

Silva da Silva, T., Martin, A., Maurer, F. and Silveira, M, S. 2011. User-Centered Design and Agile Methods: A Systematic Review. Agile Conference, Salt Lake City, 8.7-8.13, 2011, s. 77-86.

Silva da Silva, T., Silveira, M, S., Melo, Claudia. and Parzianello, L, C. 2013. Understanding the UX Designer's Role within Agile Teams. A. Marcus (Ed.): DUXU/HCII 2013, Part I, LNCS 8012, s. 599–609.

Sommerville, I. 2011. Software Engineering. 9. ed. Addison-Wesley.

Spencer, L., Ritchie, J., Lewis, J. and Dillon, L. 2003. Quality in qualitative evaluation: a framework for assessing research evidence. London: Cabinet Office. s. 1-18.

Sutherland, J. 2010. Scrum Handbook. Sommerville: Scrum Training Institute Press.

Sy, Desirée. 2007. Adapting Usability Investigations for Agile User-Centered Design. Journal of Usability Studies, 2(3), s. 112–132.

Tiainen, T. 2014. Haastattelu tietojenkäsittelytieteen tutkimuksessa. Informaatitieteiden yksikön raportteja 25/2014, Tampereen yliopisto.

Tilastokeskus. 2014. Yritystukitilasto 2012. Helsinki: Tilastokeskus. Viitattu: 10.5.2019. Osoitteesta: http://www.stat.fi/til/yrтт/2012/yrтт_2012_2014-03-25_kat_001_fi.html/

VersionOne. 2018. 12TH ANNUAL State of Agile™ Survey. Viitattu: 17.10.2018. Osoitteesta: <https://stateofagile.versionone.com/>

Williams, L. and Cockburn, A. 2003. Agile software development: it's about feedback and change. IEEE Computer Society, 36(6), s. 39-43.

Williams, H. and Ferguson, A. 2007. The UCD Perspective: Before and After Agile. Agile Conference, Washington, 13.8.-17.8., 2007, s. 285 – 290.

Womack, J. and Jones, D. 2003. Lean thinking: Banish waste and create wealth in your corporation. Simon & Schuster.

Ylöstalo, P. 2007. Keskustelua suomalaisen työelämän luonteesta ja sen muuttumisesta. Eläketurvakeskus, Helsinki.

9. Liiteluettelo

Liite 1: Kirjallisuuskatsaukseen valittu aineisto

Tutkimus	Tietokanta	Tarkoitus	Menetelmä	Keskeiset tulokset	Laatu
Anwar, S., Motla, Y. H., Siddiq, Y., Asghar, S., Hassan, M. S. and I.Khan, Z. 2014. User-centered design practices in scrum development process: A distinctive advantage?	IEEE	Ketterän käyttäjäkeskeisen suunnittelun käytäntöjen selvittäminen, jotka helpottavat integraatio haasteiden yli pääsemisessä.	Tapaustutkimus	Integraatio toi mukanaan paremman käytettävyyden, mikä puolestaan auttoi projektin onnistumista. Vaatimusten uudelleen käsittely vähensi käytettävyydestä tautsien ansiosta.	9/10
Brhel, M., Meth, H., Maedche, A. and Werdt, K. 2015. Exploring principles of user-centered agile software development: A literature review.	Science-Direct	Ketterän käyttäjäkeskeisen suunnittelun nykytilanteen selvittäminen ja käytettyjen käytänteiden kuvaaminen.	Kirjallisuuskatsaus	Tunnistettu ja analysoitu 84 oleellista julkaisua, joiden pohjalta tunnistettiin viisi käytäntöä ketterässä käyttäjäkeskeisessä suunnittelussa: tuotteen löytäminen ja tuotteen kehittäminen tulee erottaa, iteraatiivinen ja inkrementaalinen kehitys, rinnakkainen kehitys, sidosryhmien mukana pitäminen ja välinepohjainen kommunikointi.	9/10
Chamberlain, S., Sharp, H. and Maiden, N. 2006. Towards a Framework for Integrating Agile Development and User-Centred Design.	Springer	Tavoitteena luoda viitekehys ketterälle käyttäjäkeskeiselle suunnittelulle.	Tapaustutkimus	Viisi menetelmää onnistuneelle integraatiolle, jotka painottivat käyttäjän mukana pitämistä, yhteistyötä, prototyyppien tekemistä, projektinkulkua ja projektinhallintaa.	8/10
Fox, D., Sillito, J. and Maurer, F. 2008. Agile Methods and User-Centered Design: How These Two Methodologies Are Being Successfully Integrated in Industry.	IEEE	Selvittää ketterän käyttäjäkeskeisen suunnittelun toteuttamisen lähestymistavat.	Grounded theory, Haastattelut	Kolme eri lähestymistapaa integraation toteuttamiseksi löydettiin: generalisti, spesialisti ja hybridi.	9/10
Jurca, G., Hellmann, T. and Maurer, F. 2014. Integrating Agile and User-Centered Design A Systematic Mapping and Review of Evaluation and Validation Studies of Agile-UX.	IEEE	Miltä ketterä käyttäjäkeskeinen suunnittelu näyttää nykyään keskeisten julkaisujen pohjalta.	Kirjallisuuskatsaus	Isoimmaksi ongelmaksi muodostui suunnittelijan liiallinen työtaakka ja jakaminen liian moneen projektiin. Toimiviksi menetelmiksi osoittautui: työpaikat, haastattelut, skenaariot ja alhaisentason prototyyppit.	8/10
Najafi, M. and Toyoshiba, L. 2008. Two Case Studies of User Experience Design and Agile Development.	IEEE	Käyttäjakeskeisen suunnittelun integroiminen ketterään kehitykseen tapaustutkimuksien kautta.	Tapaustutkimus	Käyttäjakeskeinen suunnittelu voi myöhästyttää projektin julkaisua, mutta sen tarjoamat hyödyt hyvittävät tämän. Oleellista suunnittelijoiden ja kehittäjien rishtiin työskentely.	9/10

Salah, D., Paige, R. F. and Cairns, P., A. 2014. A systematic literature review for agile development processes and user centred design integration.	ACM	Etsiä asioita, jotka hankaloittavat ketteryyden ja käyttäjakeskeisen suunnittelun integraatiota sekä tarjota ratkaisuja näiden ylitämiseen.	Kirjallisuuskatsaus	Tutkimuksessa havaittiin hankaloittaviksi tekijöiksi seuraavat: etukäteissuunnittelun ajan riittämättömyys, suunnittelun osittaminen, suunnittelun priorisointi, työn optimointi suunnittelijoiden ja kehittäjien välillä ja suunnittelijoiden työtaakka.	8/10
Silva da Silva, T., Martin, A., Maurer, F. and Silveira, M. S. 2011. User-Centered Design and Agile Methods: A Systematic Review.	IEEE	Selvittää ketterän käyttäjakeskeisen suunnittelun toistuvat teemat ja menetelmät kirjallisuuden perusteella,	Kirjallisuuskatsaus	Tärkeimmät menetelmät integraatiossa: etukäteissuunnittelu, prototyypit, käyttäjätarinat, käyttäjätestaukset, arvioinnin tarkastus ja sprintti edellä. Tutkimustuloksena syntyi myös integraation viitekehys.	9/10
Silva da Silva, T., Silveira, M. S., Melo, Claudia. and Parzianello, L. C. 2013. Understanding the UX Designer's Role within Agile Teams.	Springer	Suunnittelijan roolien selvittäminen ketterässä käyttäjakeskeisessä suunnittelussa.	Etnografinen tutkimus	Kolme suunnittelijaroolia havaittiin integraatiossa: käyttäjäkokemussuunnittelija, interaktiosuunnittelija ja käyttöliittymäsuunnittelija.	9/10
Sy, Desirée. 2007. Adapting Usability Investigations for Agile User-Centered Design.	ACM	Käyttäjakeskeisen suunnittelun integroiminen ketterään kehitykseen organisaatiossa.	Tapaustutkimus	Käyttäjakeskeiset menetelmät tuottivat parempia suunnitteluratkaisuja kuin vesiputousmallilla tehdyt.	9/10
Williams, H. and Ferguson, A. 2007. The UCD Perspective: Before and After Agile.	IEEE	Kuinka kahta suunnittelijaa voi käyttää ketterässä tiimissä.	Tapaustutkimus	Käyttäjät pääsivät vaikuttamaan enemmän tuotteen kehitykseen. Yksi suunnittelija on vastuussa käyttäjätutkimuksesta ja toinen prototyypeistä.	8/10

Liite 2: Tutkimuksen laadun arviointi

Nr	Kysymys	Kyllä	Ei
1	Ovatko tutkimustulokset uskottavia?		
2	Lisääkö tutkimus ymmärrystä tutkittavana olevasta aiheesta?		
3	Vastaako tutkimus tarkoitukseensa?		
4	Onko aineisto kerätty kattavasti?		
5	Onko aineiston keruu menetelmät pätevät?		
6	Onko aineisto analysoitu riittävän tarkasti?		
7	Onko tutkimus selkeä ja yhtenäinen?		
8	Onko teoriatausta esitetty riittävän kattavasti?		
9	Onko tutkimusetiikkaan kiinnitetty huomiota?		
10	Onko tutkimusprosessi dokumentoitu riittävän kattavasti?		

Laaduntarkistuslista on kirjoitettu Spencer et al. [2003] tarjoaman laadunhallintaviitekehysten pohjalta. Maksimipistemäärä on 10, joka tarkoittaa myös laadukkainta. Kysymyksen vastataan kyllä/ei, joista ”kyllä” vastauksella saa yhden pisteen.

Liite 3: Haastattelukutsu

Otsikko: Haastattelukutsu

Hei (työntekijän etunimi),

Pyydän teitä haastateltavaksi Pro Gradu -tutkielmani tutkimukseen koskien käyttäjäkeskeisen suunnittelun integroimista ketterään kehitykseen. Olen viimeisen vuoden opiskelija Tampereen yliopiston tietojenkäsittelytieteissä ja tutkimukseni avulla pyritään selvittämään, miten käyttäjäkeskeinen suunnittelu integroidaan ketterään kehitykseen onnistuneesti. Lisäksi selvitetään, miten sitä sovelletaan ja käytetään suomalaisissa ohjelmistoyrityksissä.

Etsin haastateltaviksi tutkielmaani suunnittelijoita, jotka ovat työskennelleet mielellään ainakin vuoden käyttäjäkeskeisen suunnittelun parissa ketterissä ohjelmistoprojekteissa. Näkökulmanne ja ajatuksenne olisi tutkimukseni kannalta erityisen arvokkaita. Haastattelun avulla pääsette myös osallistumaan tieteelliseen tutkimukseen ja auttamaan minua eteenpäin opinnoissani. Aineisto käsitellään luottamuksellisesti ja kaikki haastateltavien henkilötiedot pidetään salassa, eikä yksittäistä haastateltavaa voi tunnistaa valmiista tutkielmasta. Yrityksen nimiä ei myöskään mainita tutkimuksessa.

Olisiko teillä mahdollisuutta osallistua tutkimukseni haastatteluun? Haastattelu-aikaa voitte ehdottaa vapaasti päivästä tai ajasta riippumatta välillä 14.1.2019 – 3.3.2019. Haastattelun kesto on noin 45min – 60min. Haastattelut voidaan toteuttaa yrityksenne tiloissa, jos mahdollista tai Tampereen yliopistolla. Millainen aika ja paikka sopisi teille?

Kaikissa tutkimukseen koskevilla kysymyksillä ottakaa vapaasti yhteyttä minuun. Ilmoitathan myös, jos koette, että jokin toinen henkilö yrityksessänne olisi sopivampi henkilö osallistumaan kyseiseen tutkimukseen.

Ystävällisin terveisin,

Joona Hirvasmaa

Tietojenkäsittelytieteiden pääaineopiskelija, Tampereen Yliopisto

Liite 4: Haastattelurunko teemoineen

Aloitus: Kiitos vielä, että suostuitte haastatteluun. Tutkimuksen avulla pyritään selvittämään, miten käyttäjäkeskeisen suunnittelun integroidaan ketterään kehitykseen onnistuneesti. Lisäksi selvitetään, miten sitä sovelletaan ja käytetään yrityksissä. Myös mallien yhdistelystä koituvia ongelmia pyritään selvittämään ja kuinka ne voidaan minimoida. Haastattelu kestää arviolta 45-60 minuuttia ja se nauhoitetaan, käykö tämä? Onko teillä vielä jotain kysyttävää? Aloitetaan haastattelu.

Taustatiedot

- Mitä työnkuvanne pitää sisällään?
- Kuinka kauan olette työskennelleet nykyisessä tehtävässänne?
- Kuuluuko työnkuviinne käyttäjäkeskeinen suunnittelu ketterässä ympäristössä?
- Yrityksen työntekijöiden määrä?
- Mitä ketterää menetelmää käytätte?

Integroitu prosessi

1. Miten käyttäjäkeskeinen suunnittelu sulautuu ketterän kehityksen prosessiin mielestäsi?
2. Miten suunnittelu- ja kehityspolut etenevät projektissa ja iteraatioissa?
3. Miten suunnittelijat ja kehittäjät kommunikoivat prosessissa?
4. Miten kokonaiskuva pidetään mielessä?

Suunnittelija ja kehitystiimi

5. Kerro millainen kehitystiimi teillä on tyypillisesti projektissa? Kuinka monta suunnittelijaa?
6. Miten suunnittelijat ja kehittäjät kommunikoivat keskenään suunnitteluratkaisuista ja suunnittelutavoitteista?
7. Millaiset tilat kehitystiimillä on?
8. Kuinka paljon aikaa suunnittelijalla on suorittaa suunnittelutoimet?
9. Millaisia suunnittelijoita yhdessä projektissa yleensä on? Roolit?
10. Monen projektin suunnittelutoimista suunnittelija vastaa?
11. Kuinka kauan suunnittelija on mukana projektin aikana?
12. Mikä vie eniten suunnittelijan aikaa projektissa?

Etukäteen suunnitteleminen

13. Milloin ensimmäiset suunnittelutoimet aloitetaan projektissa? Mitä suunnitellaan?
14. Kuinka kauan esivaihe kestää?

15. Mitä menetelmiä esivaiheessa käytetään?
16. Milloin varsinaiset ohjelmiston kehittämistyöt (koodaaminen) aloitetaan suhteessa suunnitteluun?
17. Miten suunnittelua toteutetaan esivaiheen jälkeen projektissa?
18. Miten ja koska suunnittelijat tarkastavat kehittäjien tuotoksia?

Toimintatavat ja välineet

19. Mitä toimintatapoja ja välineitä käyttäjäkeskeisessä suunnittelussa käytetään? ja miksi?
20. Miten prototyyppien käyttö näkyy projekteissa?
21. Mitkä kaksi toimintatapaa ja välinettä nostaisit tärkeimmiksi ja miksi?

Sidosryhmien roolit

22. Mikä on loppukäyttäjänrooli ohjelmiston testaamisessa?
23. Miten loppukäyttäjää pidetään mukana projektissa?

Haastattelun lopetus: Kiitos haastattelusta. Jos haluatte voin toimittaa teille graduni luettavaksi vielä ennen sen palautusta.